AÑO II - NUM. 52

# MICROHOBBY SEMANAL

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

AÑO II - N.º 52

125 PTS.
Canarias 135 ptas.

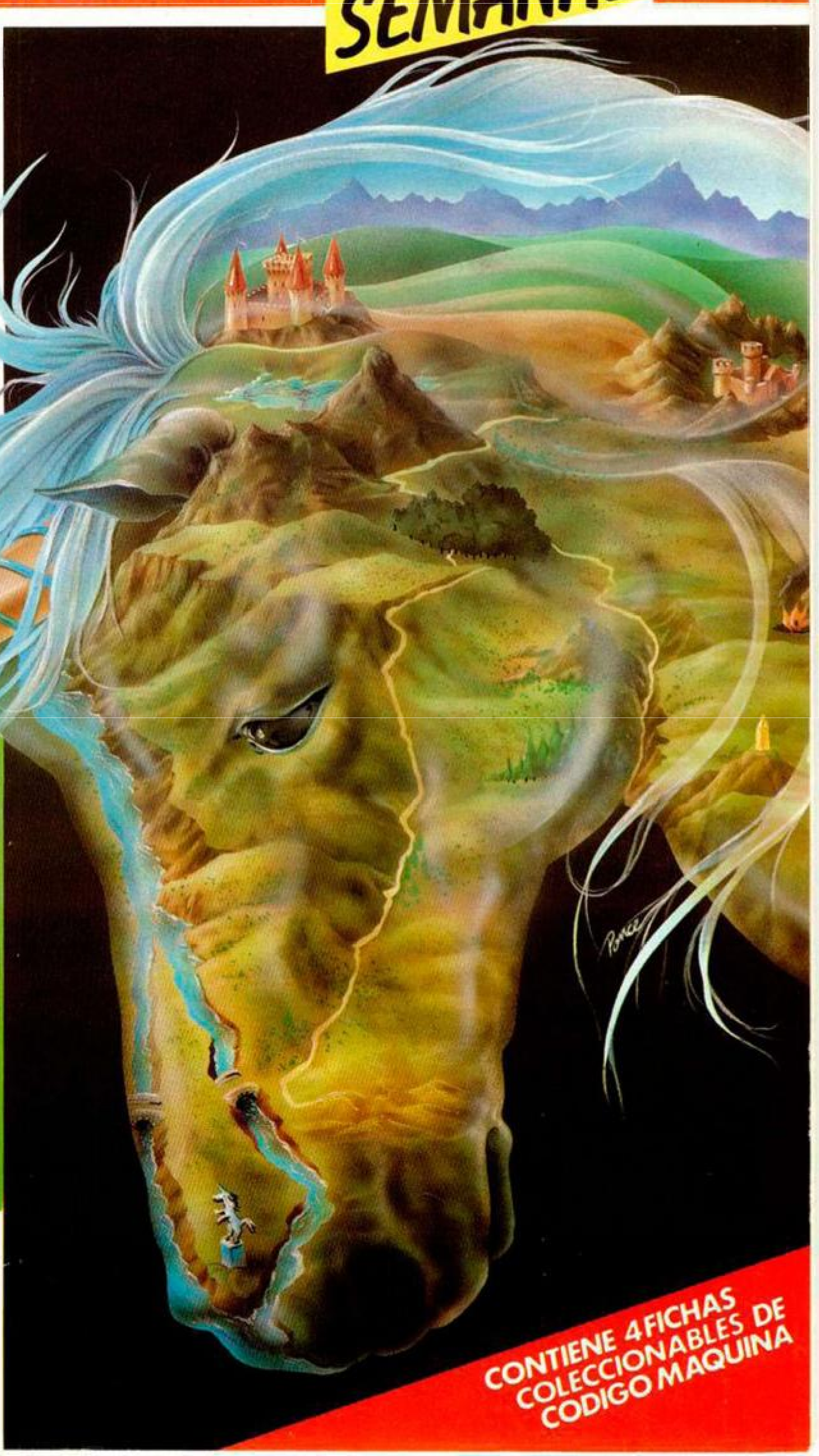
HOP EDITA HOBBY PRESS, S.A.

*PROFESOR PARTICULAR*

## CALCULO DE MOVIMIENTOS PARABOLICOS

*UTILIDADES*

## COMO COMPILAR PANTALLAS DE TEXTO



## "SHADOW OF THE UNICORN"

LA LUCHA CONTRA LAS FUERZAS DEL MAL

*MICROFILE*

## ALMACENAMIENTO DE PROGRAMAS EN MICRODRIVE

CONTIENE 4 FICHAS COLECCIONABLES DE CODIGO MAQUINA

# COMPUTIQUE

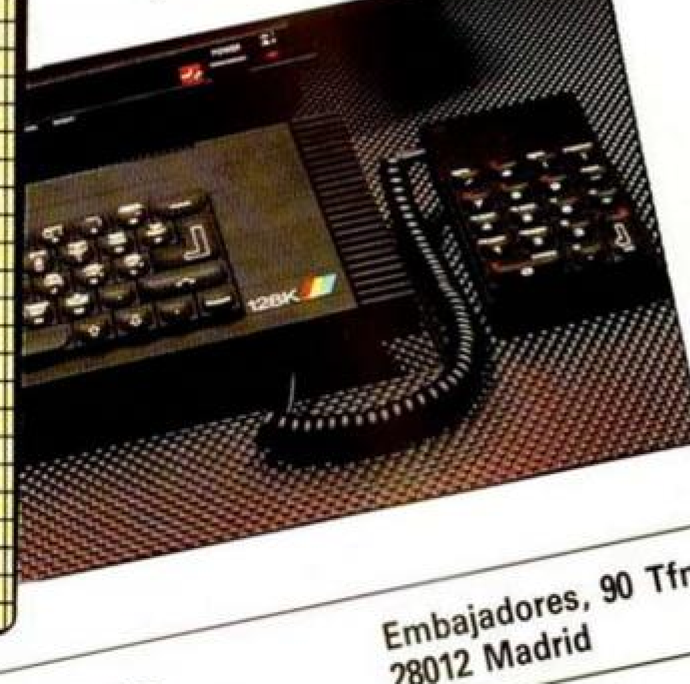
*Te da más*

GARANTIA investronica




VENTAS A PLAZOS

**¡NOVEDAD!**
**Spectrum 128 K IIIII más memoria, dos teclados y en castellano.**
**Compatible con Spectrum 48 K/Plus**
***¡Más de 5.000 programas!***




**Al comprar tu spectrum**
*te regalamos*

- Joystick Gunshot I
- Estuche con seis juegos originales
  - Backgammon
  - Reversi
  - Planetoids
  - Cookie
  - Psst
  - Jet Pac
- Un estupendo libro de Basic para niños
- Y además te obsequiamos con un curso de introducción al Basic
- El mejor programa «Pole Position»

Embajadores, 90 Tfno. 2270980
28012 Madrid

Servimos a tiendas
Abrimos sábados por la tarde

COMPUTIQUE

**Director Editorial**
José I. Gómez-Centurión

**Director Ejecutivo**
Domingo Gómez

**Asesor técnico editorial**
Gabriel Nieto

**Redactor Jefe**
Africa Pérez Tolosa

**Diseño**
Rosa María Capitel

**Redacción**
Amalio Gómez,
Pedro Pérez
Jesús Alonso

**Secretaria Redacción**
Carmen Santamaría

**Colaboradores**
Primitivo de Francisco,
Rafael Prades,
Miguel Sepúlveda,
Sergio Martínez y J. M. Lazo

**Fotografía**
Javier Martínez, Carlos Candel

**Portada**
José María Ponce

**Dibujos**
J. R. Ballesteros, A. Perera,
F. L. Frontán, Pejo, J. M. López
Moreno J. Igual, J. A. Calvo, Loriga,
J. Olivares

**Edita**
HOBBY PRESS, S. A.

**Presidente**
María Andrino

**Consejero Delegado**
José I. Gómez-Centurión

**Jefe de Publicidad**
Marisa Esteban

**Publicidad Barcelona**
José Galán Cortés
Tels.: 303 10 22 - 313 71 76

**Secretaria de Dirección**
Marisa Cogorro

**Suscripciones**
M.ª Rosa González
M.ª del Mar Calzada

**Redacción, Administración y Publicidad**
La Granja, s/n
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR

**Dto. Circulación**
Carlos Peropadre

**Distribución**
Coedis, S. A. Valencia, 245
Barcelona

**Imprime**
Rotedic, S. A. Ctra. de Irun,
km. 12,450 (MADRID)

**Fotocomposición**
Espacio y Punto, S. A.
Paseo de la Castellana, 268

**Fotomecánica**
Grof
Ezequiel Solana, 16

**Depósito Legal**
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cia. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).





Solicitado control
OJD

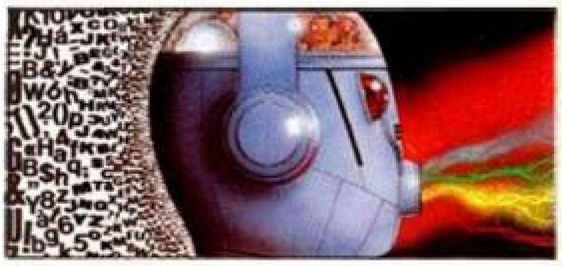

# MICROHOBBY ESTA SEMANA


**AÑO II. N.º 52. 12 al 18 de noviembre de 1985**
**125 ptas. (Sobretasa Canarias 10 ptas.)**









***RUTINA DE UTILIDAD***
***Para la compilación de textos***
***Pág. 26***

# MICROHOBBY NUMEROS ATRASADOS

Queremos poner en conocimiento de nuestros lectores que para conseguir números atrasados de MICROHOBBY SEMANAL, no tienen más que escribirnos indicándonos en sus cartas el número deseado y la forma de pago elegida de entre las tres modalidades que explicamos a continuación.
Una vez tramitado esto, recibirá en su casa el número solicitado por el precio de 95 ptas., cada número, más 25 ptas. por gastos de envío.

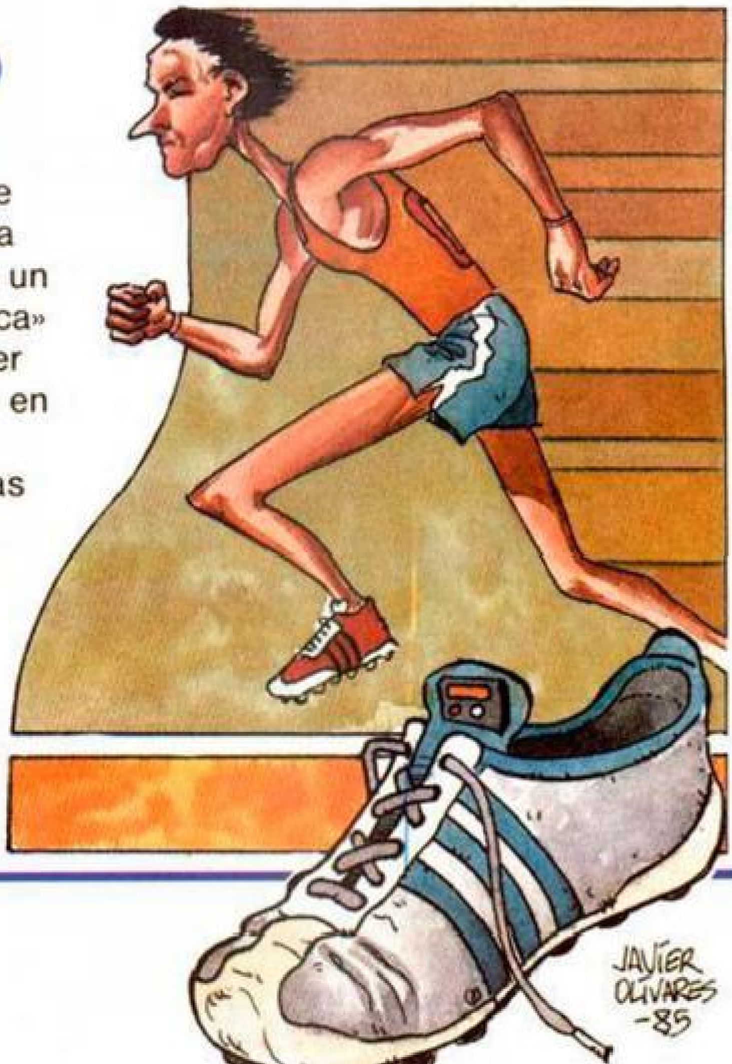



## FORMAS DE PAGO

- Enviando talón bancario nominativo a Hobby Press, S. A. al apartado de Correos 54062 de Madrid.
- Mediante Giro Postal, indicando número y fecha del mismo.
- Con Tarjeta de Crédito (VISA o MASTER CHARGE), haciendo constar su número y fecha de caducidad.

## CON LOS PIES EN EL FUTURO

No se trata de un proyecto futurista. Tampoco de una broma. Lo cierto es que la conocida marca de zapatillas Adidas, acaba de idear una, denominada Micro Pacer, que lleva incorporada en la lengüeta un ordenador. Gracias a este artilugio y por la «módica» cantidad de 25.000 ptas., el usuario podrá conocer a la perfección la distancia recorrida en su paseo en kilómetros, el tiempo por distancia (en horas), la velocidad media alcanzada, el consumo de calorías por trayecto y, por si esto era poco, una unidad de memoria individualmente ajustable permite determinar la distancia de carrera total y su duración durante un largo período de tiempo.
Este maravilloso invento estará a la venta en la próxima primavera.




La última creación Ocean

## THE NEVER ENDING STORY

Un despliegue de imaginación y aventura se han dado la mano en la última creación de OCEAN, The Never Ending Story, en donde monstruos y seres irreales configuran una emocionante historia basada en el libro de Michael Ende «La historia interminable».

Caracterizado como el héroe Atreyu y ayudado por Falcor el «dragón de la suerte», te enfrentarás con infinidad de pruebas y terrores, penetrando en una trama de imaginación y fantasía a tope.




Creemos que son suficientes alicientes como para pensar que se trata de algo verdaderamente recomendable que esperamos con impaciencia. Mientras tanto, sólo sabemos que la cinta está en el mercado inglés para Commodore, Amstrad y Spectrum.

Realizada por la Generalitat

## GUIA DE PROGRAMAS PARA ORDENADOR DOMESTICO

El Centro divulgador de la informática, organismo dependiente de la Generalitat de Catalunya, acaba de anunciar la pronta aparición de una guía que contendrá información directa acerca de unos 500 programas orientados al ordenador familiar y escolar, cuya primera edición constará de 14.000 ejemplares (7.000 en castellano y otros tantos en catalán).

La idea resulta bastante interesante, pues esta guía incluirá además la dicha información sobre todo tipo de programas (base de datos, tratamiento de textos, juegos, simuladores, utilidades, etc.) una numerosa y variada información referente a todos los temas que en la actualidad configuran el mundo de la informática, como pueden ser una relación de las casas de Software en España, distribuidoras, revistas especializadas existentes en nuestro país y en todo el mundo, y otros muchos apartados que, sin duda alguna, serán de mucha utilidad para todas aquellas personas interesadas en el tema de la programación.

# MICRO HITS

1. **PROFANATION** *(Dinamic)*
2. **KNIGHT LORE** *(ULtimate)*
3. **UNDERWURLDE** *(Ultimate)*
4. **EVERYONE'S A WALLY** *(Mikro Gen)*
5. **ALIEN 8** *(Ultimate)*
6. **SKOOL DAZE** *(Microsphere)*
7. **MATCH POINT** *(Psion)*
8. **SPY HUNTER** *(U.S. Gold)*
9. **MATCH DAY** *(Ocean)*
10. **BASEBALL** *(Imagine)*
11. **HYPERSPORTS** *(Imagine)*
12. **JET SET WILLY** *(S. Proyects)*
13. **ROCKY HORROR SHOW** *(C.R.L.)*
14. **BRUCE LEE** *(U.S. Gold)*
15. **ROCKY** *(Dinamic)*
16. **TIR NA NOG** *(Gargoyle Games)*
17. **DUN DARACH** *(Gargoyle Games)*
18. **DECATHLON** *(Ocean)*
19. **PYJAMARAMA** *(Mikro Gen)*
20. **RAID OVER MOSCOW** *(U.S. Gold)*





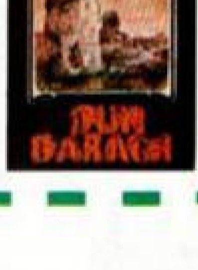






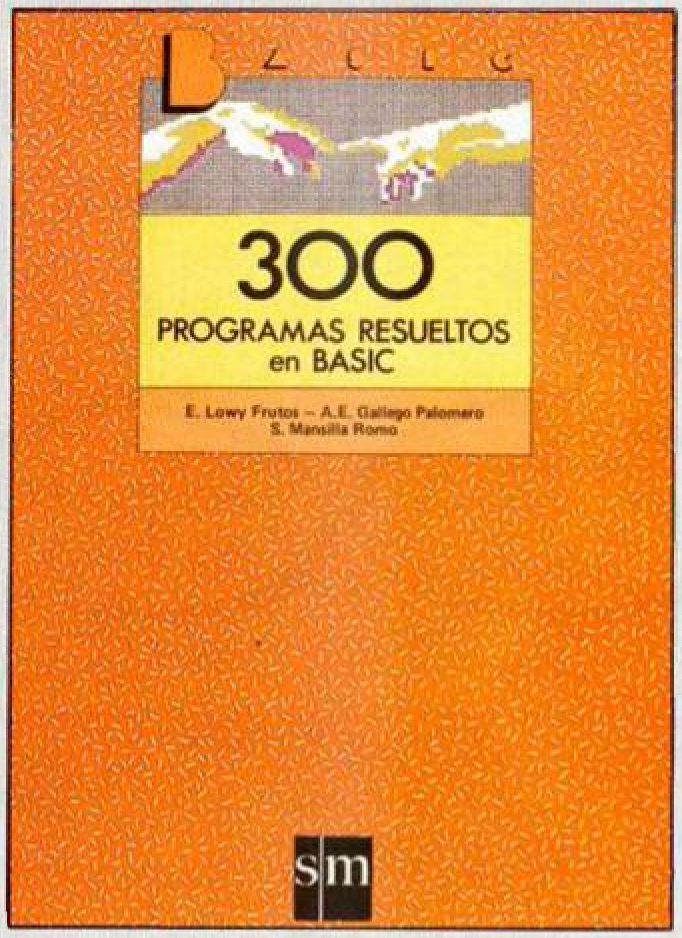







Os recordamos a todos los lectores que podéis votar por carta o por teléfono.

# LIBROS




## 300 PROGRAMAS RESUELTOS EN BASIC

Lowy Frutos/Gallego Palomero/Mansilla Romo

Ediciones S. M. 190 páginas.

Como complemento a la anterior publicación del libro BASIC Programación, la editorial S. M. ha realizado un nuevo libro enmarcado dentro de la colección Basic, cuyo contenido, como bien indica el título, consta de un gran número de programas en Basic con sus soluciones correspondientes.

El libro comienza con una breve introducción que explica qué es un microprocesador y sus partes fundamentales, así como el procedimiento para cargar y ejecutar un programa.

El resto de los capítulos (14 en total), presentan idéntica estructura y constan de los siguientes apartados:

— Resumen de las instrucciones. En él se detallan cada una de las instrucciones que se van a utilizar en los programas del capítulo, además de incluir algunos ejemplos ilustrativos.

— Programas resueltos explicados. Contiene un buen número de programas cortos que dan respuesta a otras tantas situaciones concretas y que vienen convenientemente acompañados de una serie de comentarios que facilitan enormemente la comprensión de los mismos.

— Programas resueltos. Más programas con sus correspondientes soluciones, pero esta vez sin comentarios, puesto que se supone que el lector ya ha asimilado en una buena medida, el funcionamiento de la instrucción tratada en el correspondiente capítulo.

En definitiva, es un libro que puede servir de mucha ayuda para los principiantes en la programación en Basic y con el que seguramente acabarán dominando dicho lenguaje.

# TRUCOS



## SCROLL LATERAL

Si queremos conseguir una simulación de scroll lateral, Joan Tarruel nos recomienda una rutina en la que aparecerá el scroll como si fuera visto por una ventana, pero jugando con las variables y los bucles, se puede conseguir cualquier otra cosa.

```
  10 PRINT PAPER 4;AT 12,11;"
 "
  20 GO TO 100
  30 FOR n=1 TO 4
  40 READ a
  50 PRINT PAPER 4;AT 12,10+n;CH
R$ a
  70 NEXT n
  80 RETURN
 100 GO SUB 30
 110 FOR m=1 TO 23
 120 RESTORE
 130 FOR x=1 TO m
 140 READ a
 150 NEXT x
 160 GO SUB 30
 170 NEXT m
 180 RESTORE
 190 GO TO 10
 200 DATA 32,32,32,32,77,73,67,8
2,79,72,79,66,66,89,32,32,115,10
1,109,97,110,97,108,32,32,32,32
```

## ENTRE TITULOS ANDA EL TEMA

José Santiago García nos ha mandado una serie de trucos encaminados a mejorar o variar los títulos de nuestros programas. Ahí van:

Para rotular boca abajo:

```
  10 INPUT A$: REM A$ NO MAS DE
 31 CARACTERES
  20 PRINT AT 21,0; INK 7;A$
  30 FOR X=0 TO (LEN A$)*8: FOR
Y=0 TO 100
  35 IF POINT (X,Y)=1 THEN PLOT
X,100-Y
  40 NEXT Y: NEXT X
```

Para hacer carteles rotatorios:

```
  10 INPUT A$
  20 FOR Y=1 TO LEN A$: FOR N=30
 TO 1 STEP -1
  30 POKE 23606,N: PRINT AT 3,Y;
A$(Y TO Y)
  40 NEXT N: NEXT Y
```

Para hacer rótulos grandes:

```
  10 INPUT A$: PRINT AT 21,0; IN
K 7;A$
  20 FOR Y=0 TO 7: FOR X=0 TO (L
EN A$)*8
  21 IF POINT (X,Y)=1 THEN PLOT
X*2,100+Y*2: DRAW 2,0: DRAW 0,2:
 DRAW -2,0: DRAW 0,-2: BEEP .004
,50
  22 NEXT X: NEXT Y
```

## EDITAR EL "INKEY$"

Esta rutina que os ofrecemos a continuación, de José Luis, permite que el Spectrum reconozca la tecla pulsada sin necesidad de utilizar el comando INKEY$.

Como todos sabemos, el inconveniente de este comando es el poder pulsar solamente una tecla, con lo que en nuestros juegos no hay posibilidad de movimiento en ocho direcciones sin que ello resulte demasiado engorroso (ya que deberíamos controlar ocho teclas).

```
 1 LET a=1
 2 LET B=1
 3 LET c= IN 63486
 4 IF c=191 THEN LET c=190
 5 IF c=255 THEN LET c=254
10 IF IN 64510=c THEN LET a=a-1
20 IF IN 57342=c THEN LET b=b+1
30 IF IN 65022=c THEN LET a=a+1
40 IF IN 61438=c THEN LET b=b-1
45 PRINT AT a+1,b;"   "
50 PRINT AT a,b;" X "
60 PRINT AT a-1,b"   "
70 GO TO 10
```

La solución a este problema la encontramos en el comando "IN" ya que éste reconoce la tecla pulsada, permitiendo, además, pulsar varias a la vez.

Existen en el mercado varias series de Spectrum para las cuales la función "IN" es diferente.

En las versiones 2 y 3 B el valor de esta función es 64 más que en la versión 3. Pero para ver esto con más detalle basta hacer PRINT IN 63486; si es de versión 2 ó 3 B obtendremos el valor 255, mientras que la otra versión nos dará 191.

En este programa (líneas 3, 4 y 5) nuestro ordenador calcula el valor de IN 63486, y se corrige automáticamente tanto si es de una u otra versión.

Las teclas a emplear son:

- Q Arriba
- A Abajo
- O Izquierda
- P Derecha

# TANDEM LUNAR

Juan Carlos HUSILLOS

**Spectrum 48 K**

TODAS LA LETRAS MAYUSCULAS SUBRAYADAS DEBERAN TECLEARSE EN MODO GRAFICO.

**Una expedición a la luna con tres astronautas ha tenido problemas y su comunicación con la tierra ha quedado cortada. Temiéndonos lo peor, nuestra nave de socorro se dispone a ir en su ayuda.**

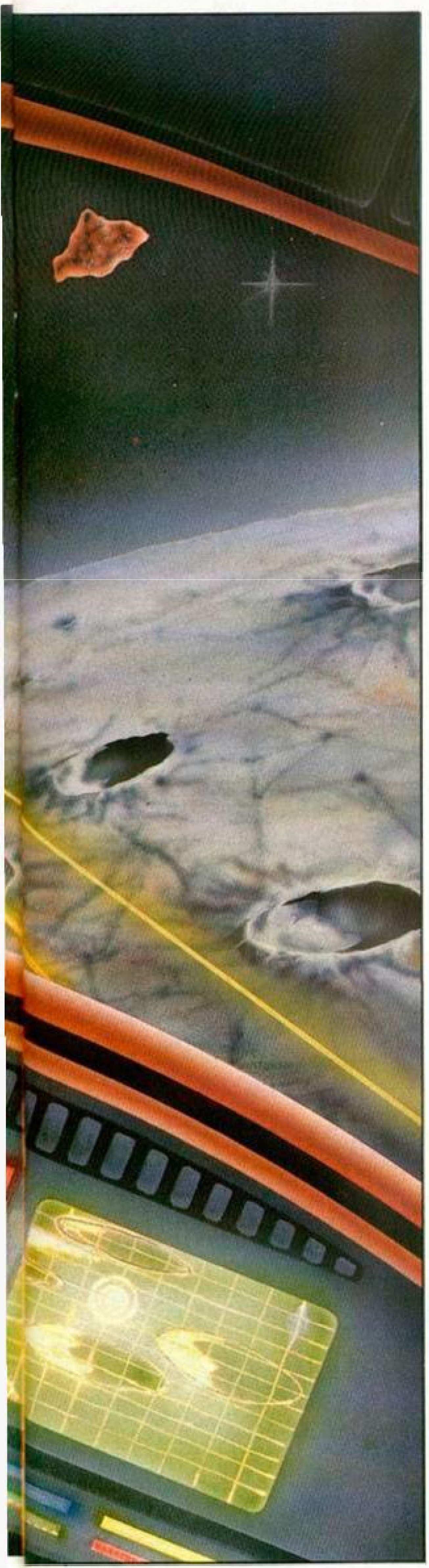
Esquivando una gran barrera de meteoritos que rodean la esfera lunar, tendremos que conseguir aterrizar en su superficie y, concretamente en las plataformas previstas para ello, evitando, al mismo tiempo, el ataque de sus habitantes que intentarán cortarnos el paso. Pero otra dificultad viene a unirse a nuestra arriesgada misión: debido a la poca capacidad de la nave de salvamento sólo podremos recoger a un astronauta por lo que nos veremos obligados a realizar tres viajes.

Si superamos esta triple hazaña, nos daremos por satisfechos.






```
   1 GO SUB 9000: LET mu=0: LET pu=0: LET vi=3: PAPER 0: INK 6: BORDER 3: CLS : GO SUB 1000
   2 LET fu=f1: DIM b$(32+10): LET b$(31 TO 62)="G    G  G G  G   G G  GG"
   3 LET b$(155 TO 186)="GG    GG  G      GG    G      G"
   4 LET b$(217 TO 248)="G       G  GG   GGGG      G G     GG"
   5: LET y=1: LET x1=14: GO SUB 100: LET x=n+2
   6 LET b$=b$(310)+b$(1 TO 309)
   8 IF INKEY$="p" THEN PRINT AT y,x;" ": LET x=x+1: PRINT AT y,x;"A"
   9 IF INKEY$="o" THEN PRINT AT y,x;" ": LET x=x-1: PRINT AT y,x;"A"
  10 IF INKEY$="q" AND fu>0 THEN LET fu=fu-5: LET y=y-1: PRINT AT y+2,x;"I"
  11 PRINT AT y,x;" ": LET y=y+1: PRINT INK 7;AT y,x;"B": IF y>=19 AND (x=5 OR x=18 OR x=27) AND ATTR (y+1,x)=3 THEN GO TO 2000
  12 BEEP .012,20+y: PRINT AT 3,0;b$: PRINT #1; INK 0; PAPER 3; BRIGHT 1;AT 0,7;PU;AT 0,19;fu;" "
  13 IF SCREEN$ (y,x)<>" " AND y>=3 AND y<=12 THEN GO TO 3000
  14 IF ATTR (y,x+1)<>6 OR ATTR (y,x-1)<>6 THEN GO TO 3000
  15 GO TO 6
 100 FOR n=0 TO 26: PRINT AT 0,n;" BCD ";AT 1,n;" EAF "
 110 IF INKEY$=" " THEN RETURN
 115 LET b$=b$(310)+b$(1 TO 309)
 116 PRINT AT 3,0;b$
 120 NEXT n: FOR n=26 TO 0 STEP -1: PRINT AT 0,n;" BCD ";AT 1,n;" EAF "
 121 LET b$=b$(310)+b$(1 TO 309)
 122 PRINT AT 3,0;b$
 130 IF INKEY$=" " THEN RETURN
 140 NEXT n: GO TO 100
1000 INK 7: PLOT 0,0: DRAW 8,8: DRAW 8,-8: DRAW 16,8: PLOT 48,8: DRAW 16,15: DRAW 16,-15: DRAW 32,40: DRAW 8,-20: DRAW 16,-20: PLOT 152,8: DRAW 32,10: DRAW 8,-5: DRAW 8,10: DRAW 8,-20: PLOT 224,0: DRAW 31,10
1010 PRINT AT 21,4;"█";AT 21,17;"█"; INK 3;AT 20,5;"C";AT 20,18;"C"
1020 PRINT AT 21,26;"█"; INK 3;AT 20,27;"C"
1030 PLOT 0,0: DRAW 255,0: PRINT #1;AT 0,0; PAPER 3; BRIGHT 1; INK 0;"PUNTOS=        FUEL=    NAVES AAA"
1039 INK 6
```



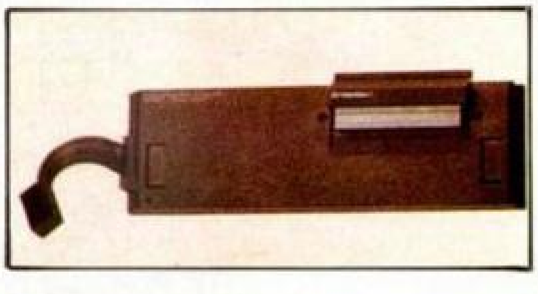
```
1040 RETURN
2000 LET ki=x: LET y1=20: DIM b$
(320): LET a$="M     M     M
 M     M      ": LET c$="
                          ": LET
b$(64 TO )=a$+c$+a$+c$+a$+c$+a$
2001 FOR j=1 TO x: LET pu=pu+5:
PAUSE 5: PRINT AT 21,j; OVER 1;"
J";AT 21,j-1;"J": BEEP .03,j: NE
XT j: PRINT AT 21,j-1; OVER 1;"J
": PAUSE 20
2009 FOR k=19 TO 1 STEP -.25: IF
 y1>=19 THEN PRINT AT y1,ki;" ";
AT y1-2,ki-1;" ": LET ki=INT (RN
D*31): LET y1=8
2010 LET b$=b$(310)+b$(1 TO 309)
2011 IF y1<19 THEN PRINT AT y1,k
i;" ";AT y1-2,ki-1;" ": LET y1=y
1+1: PRINT AT y1,ki;"I";AT y1-2,
ki-1;"I": IF (y1=INT y OR y1-2=I
NT y) AND (ki=x OR ki-1=x) THEN
GO TO 3000
2012 PRINT #1; INK 0; PAPER 3; B
RIGHT 1;AT 0,7;PU;AT 0,19;fu;" "
2021 IF INKEY$="p" THEN PRINT AT
 y,x;" ": LET x=x+1: PRINT AT y,
x;"A"
2022 BEEP .012,50-k: IF INKEY$="
o" THEN PRINT AT y,x;" ": LET x=
x-1: PRINT AT y,x;"A"
2023 IF INKEY$="q" AND fu>0 THEN
 LET fu=fu-1: LET k=k-.75: PRINT
 AT y+1,x; INK 7; BRIGHT 1;"I";A
T y+1,x;" "
2024 PRINT AT y,x;" ": LET y=k:
PRINT INK 7;AT y,x;"A"
2028 PRINT AT 3,0;b$
2029 IF SCREEN$ (y,x)<>" " AND y
>=3 AND y<=12 THEN GO TO 3000
2030 IF ATTR (y,x+1)<>6 OR ATTR
(y,x-1)<>6 THEN GO TO 3000
2050 NEXT k
2060 IF x<>n+2 THEN GO TO 3000
2065 FOR h=0 TO 200: NEXT h: PAU
SE 200: PRINT AT y1,ki;" ";AT y1
-2,ki-1;" "
2069 LET mu=mu+1: LET pu=pu+fu
2070 PRINT AT 0,n;"    ";AT 1,n;
"    ": LET pu=pu+250: IF mu>=3
THEN GO TO 2090
2080 GO TO 2
2090 IF f1=50 THEN LET PU=PU+500
: PRINT AT 10,0;"SALVASTE A TODO
S LOS ASTRONAUTASPERO AHORA SERA
 MAS DIFICIL POR QUE TU NAVE SE
HA AVERIADO Y NO TIENES FUEL": L
ET F1=0: PAUSE 10: PAUSE 0: CLS
: GO SUB 1000: GO TO 2
2100 LET pu=pu+900: IF mu<>6 THE
N GO TO 2
2110 LET f1=25: GO TO 1
3000 LET mu=0: PRINT AT y,x; FLA
SH 1;"A": FOR j=0 TO 20: BEEP .0
1,0: BEEP .01,10: BEEP .01,-10:
BORDER 1: BORDER 3: NEXT j: LET
vi=vi-1: IF vi<0 THEN PRINT FLAS
H 1; BRIGHT 1; INK 7;AT 10,10;"G
AME OVER": GO TO 3500
3010 PRINT #1;AT 0,0;"PULSE UNA
TECLA                ": PAUSE 1
: PAUSE 0: CLS : GO SUB 1000: PR
INT #1;AT 0,29+vi; OVER 1; PAPER
 3; INK 0; BRIGHT 1;"X"+("X" AND
 vi(2)+("X" AND vi(1)
3020 GO TO 2
3500 IF pu>=re THEN LET re=pu: I
NPUT "CONSEGUISTE MEJORAR ESCRIB
E TU  NOMBRE",R$: IF LEN R$=12 T
HEN GO TO 3500
3510 PAUSE 1: PAUSE 200: CLS : P
RINT AT 10,0;"EL MEJOR = "; FLAS
H 1;r$''TAB 10;"CON = ";re
3520 PRINT #0;"OTRA PARTIDA S/N"
3530 IF INKEY$="s" THEN LET f1=5
0: GO TO 1
3540 IF INKEY$="n" THEN GO TO 10
000
3550 GO TO 3530
9000 REM GRAFICOS
9010 FOR N=USR "A" TO USR "j"+7:
 READ A: POKE N,A: NEXT N
9020 DATA 0,24,60,126,66,60,36,6
6,0,0,1,15,63,127,255,255,60,255
,195,255,195,255,195,255
9030 DATA 0,0,128,240,252,254,25
5,255,255,143,99,31,7,3,1,1,255,
241,198,248,224,192,128,128
9040 DATA 24,36,66,130,129,65,34
,28,0,0,24,60,90,255,165,90,24,3
2,4,66,16,133,32,0
9050 DATA 56,56,16,254,16,16,40,
68
9060 PAPER 0: INK 7: BRIGHT 1: B
ORDER 2: CLS : LET F1=50: LET re
=100: LET r$="CPU Z80"
9070 PLOT 30,100: DRAW -10,0: DR
AW 0,20: PLOT 40,120: DRAW 0,-20
: DRAW 10,0: DRAW 0,20
9080 PLOT 60,100: DRAW 0,20: DRA
W 10,-20: DRAW 0,20: PLOT 80,100
: DRAW 0,20: DRAW 10,0: DRAW 0,-
20: PLOT 80,110: DRAW 10,0
9090 PLOT 100,100: DRAW 0,20: DR
AW 10,0: DRAW 0,-10: DRAW -10,0:
 PLOT 105,110: DRAW 5,-10
9100 PLOT 105,70: DRAW 0,20: PLO
T 100,90: DRAW 10,0: PLOT 120,70
: DRAW 0,20: DRAW 10,0: DRAW 0,-
20: PLOT 120,80: DRAW 10,0
9110 PLOT 140,70: DRAW 0,20: DRA
W 10,-20: DRAW 0,20: PLOT 160,70
: DRAW 0,20: DRAW 5,0: DRAW 5,-5
: DRAW 0,-10: DRAW -5,-5: DRAW -
5,0
9120 PLOT 180,70: DRAW 0,20: DRA
W 10,0: PLOT 180,8": DRAW 10,0:
PLOT 180,70: DRAW 10,0
9130 PLOT 200,70: DRAW 0,20: DRA
W 10,-20: DRAW 0,20
9140 PRINT AT 21,0;"(C) JUAN CARL
OS MUSILLOS  1985        "
9150 PAUSE 0: CLS
9160 PRINT AT 5,0;"TRES ASTRONAU
TAS SE HAN ESTRE-  LLADO EN LA L
UNA Y TU MISION    ES RESCATARLE
S,PERO EN LA NAVE  DE SALVAMENTO
 SOLO CABE UN      ASTRONAUTA,AS
I QUE TENDRAS  QUE HACER TRES VI
AJES,"'''"PULSA UNA TECLA": PAUSE
 0: CLS
9170 PRINT AT 5,0;"PARA CONSEGUI
R ATERRIZAR PRIMEROTIENES QUE PA
SAR POR LA BARRERA DE METEORITOS
 Y LUEGO ATERRIZAR EL LAS PLATAF
ORMAS,CADA PLATAFORMA SOLO SE PU
EDE USAR 1 VEZ.    LUEGO PARA SU
BIR TENDRAS QUE    ESQUIBAR A LA
S NAVES ENEMIGAS   "
9180 PRINT AT 21,0;"PULSA UNA TE
CLA": PAUSE 0: CLS
9190 BRIGHT 0: PRINT AT 5,0;"
C  O  N  T  R  O  L  E  S   "''
'"O= IZQUIERDA"''"P= DERECHA"''"Q=
 SUBIR"''"SPACE= SALIR DE LA NAVE
 NODRIZA"''''"PULSA UNA TECLA PAR
A JUGAR": PAUSE 0: CLS : RETURN
```

## PROGRAMAS DE LECTORES

Ya tenemos al afortunado ganador de un MICRODRIVE y su INTERFACE correspondiente en el sorteo realizado entre todos los programas de los lectores recibidos en el mes de octubre.

Se trata de JULIO BARRANTES ASTUDILLO, que vive en la calle Ribadavia, 18 de Madrid.

¡Enhorabuena!





# BABETA.S.A.CO

distribuye en exclusiva para toda España

**ROMANTIC ROBOT**

para tu ZX Spectrum 48 K




NUEVO
P.V.P. 2.900 pts.

NUEVO

"TURBO-KILLER"

LO MAXIMO EN COPIAS DE SEGURIDAD

**COMPATIBLE CON SPECTRUM 128 K**

VENTA EXCLUSIVA CONTRA REEMBOLSO,
SIN GASTOS DE ENVIO. VENTA DIRECTA

*Galileo, 25 - Entreplanta A*
*Tels. 447 97 51 / 447 98 09*
*28015 MADRID*

Nombre ..........
Dirección .......... Población ..........
Código P. .......... Pedido ..........

---

P.V.P. 2.500 pts

HG

**Holding Garijo, s.a.**

NUEVO

**presenta para toda España**

## "EDITOR MUSICAL"

un programa 100% codigo máquina
que le convertirá su *Spectrum* en una Imprenta Musical
Toda la música puede ser pasada a
*DATA* con el programa *DATA CONVERTER*

**EDITOR MUSICAL + DATA CONVERTER**

**+ MICRO DRIVER: 2.500 pts.**




***Incluye plantilla para***
***el Spectrum Normal y Spectrum Plus***

*Holding Garijo, s. a.*
*DISTRIBUCION INSTRUMENTOS MUSICALES*

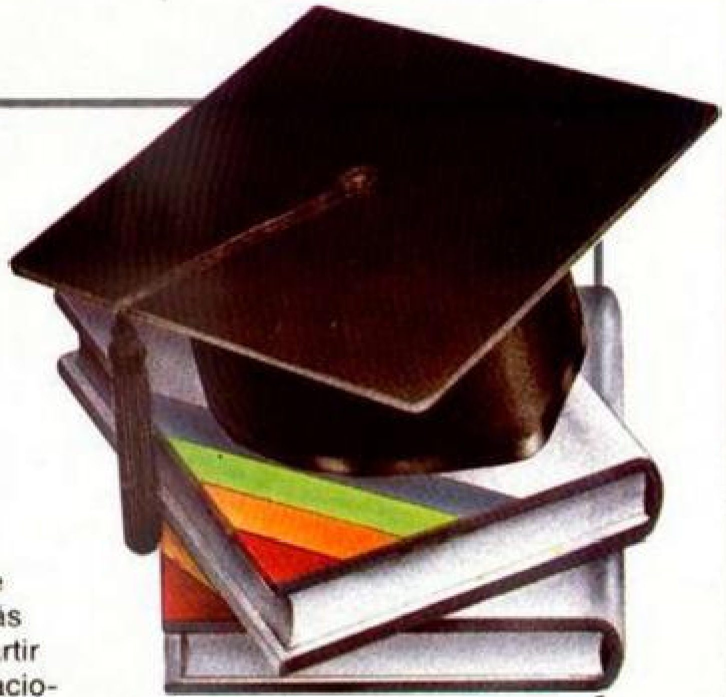
*Alonso Núñez, 28 - Nave 1.ª*
*Tel. 459 25 00 (4 líneas)*
*28039 Madrid*

Distribución en exclusiva para toda España

Nombre ..........
Dirección .......... Población ..........
Código P. .......... Pedido ..........

# Profesor particular

Muchas veces, resolver problemas de física resulta entretenido y para demostrártelo te proponemos un programa que te será de gran utilidad.



## MOVIMIENTO PARABOLICO

Dicho programa te permite las siguientes opciones:

a) Obtener las ecuaciones cinemáticas más importantes del tiro parabólico.

b) Simular el tiro con valores iniciales a tu elección.

La posibilidad de simular el tiro te proporciona una idea de los órdenes de magnitud de los valores de velocidad inicial, ángulo de tiro, alcance, tiempos característicos, etc.

c) Por último, el programa te puede proponer problemas variados de tiro parabólico, para que tú los resuelvas (empezando por ver si tienen solución y si tienen más de una).

Si lo requieres, te puede aconsejar (pulsando ayuda) sobre el método de resolución más apropiado a partir de las cinco ecuaciones del apartado a).

A propósito: Si quieres comprobar tu solución, no tienes más que ir al apartado b) y... disparar. Así no tienes ninguna duda del resultado. Los datos del problema están disponibles automáticamente pulsando "p" cuando te pidan las variables.

```
   5 CLEAR 39999
   7 PAPER 5: BORDER 5: CLS : PRINT INK 1;AT 5,5;"MOVIMIENTO PARABOLICO"
  10 FOR n=USR "a" TO USR "a"+127
  15 READ a: POKE n,a: NEXT n
  30 DATA 1,2,4,9,16,34,65,255
  32 DATA 128,64,32,144,72,68,130,255
  34 DATA 32,46,42,42,46,32,32
  36 DATA 32,4,4,4,4,244,148,148,148
  38 DATA 7,24,32,32,64,64,32,32
  40 DATA 224,24,4,4,2,2,4,4
  42 DATA 24,20,11,8,4,15,7,7
  44 DATA 24,40,208,16,32,240,224,224
  46 DATA 15,16,24,127,128,128,128,127
  48 DATA 240,8,24,254,1,1,1,254
  50 DATA 0,1,2,4,4,24,32,32
  52 DATA 16,8,4,5,9,16,16,15,0
  54 DATA 220,34,2,154,36,18,18
  56 DATA 99,1,139,2,28,8,136,112
  58 DATA 96,144,32,64,240,0,0,0
  60 DATA 0,0,59,68,68,76,55,0
  62 DIM M(5,4): FOR i=1 TO 5: FOR j=1 TO 2: READ a: LET M(i,j)=a: NEXT j: NEXT i
  64 DATA 150,10,85,5,310,30,400,50,350,100
  70 FOR i=40000 TO 40025: READ a: POKE i,a: NEXT i
  72 DATA 6,5,197,33,0,3,17,1,0,229,205,181,3,225,17,16,0,167,237,82,32,240,193,16,233,201
  73 LET r=.12345: FOR i=1 TO 5: READ a: LET M(i,4)=a: NEXT i: DATA 80,60,170,100,350
  75 INK 0: PRINT AT 10,1;"-PULSA:"
  80 PRINT AT 12,3;"1)Breve explicacion teorica.";AT 14,3;"2)Experimentacion y comproba-      cion de resultados de pro-      blemas.";AT 18,3;"3)Propuesta de problemas."
  90 IF INKEY$<>"1" AND INKEY$<>"2" AND INKEY$<>"3" THEN GO TO 90
  95 GO TO (VAL INKEY$+1)*1000
1000 INK 4: PAPER 5: BORDER 5
1010 CLS : PLOT 0,0: DRAW 60,20: DRAW 51,-12: PLOT 94,12
1020 DRAW 56,16: DRAW 60,-8: PLOT 180,16: DRAW 60,8: DRAW 15,-2
1030 DRAW 0,-22: DRAW -255,0
1040 DRAW 0,10: DRAW 18,-2
1100 PRINT AT 20,xc; INK 2;"  ";AT 21,xc; INK 1;"  "
1200 PRINT AT xg,yg; INK 3;" ■";AT xg+1,yg; INK 1;"■■"
1300 INK 0: PRINT AT 21,0;"■■": PLOT 8,7: DRAW 7,4
1420 LET dis=SIN (2*alfa)*v0*v0/9.81: LET ax=dis*255/500
1435 GO SUB 1700
1440 LET hmax=v0*v0*SIN alfa*SIN alfa/2/9.81
1445 LET jj=500/255/v0/COS alfa
1450 LET hx=hmax*255/500: LET jjj=4*hx/ax/ax: LET dado=0
1455 PRINT AT 6,1;"pulsa una tecla para disparar": PAUSE 0
1457 PRINT OVER 1;AT 6,1;"pulsa una tecla para disparar": INK 2
1460 FOR i=10 TO 255
1470 LET j=jjj*i*(ax-i)
1480 IF j>250 THEN LET dado=0: PRINT #0;"Te has pasado, tio!": RETURN
1485 LET t=i*jj: PRINT AT 2,2; PAPER 7;.01*INT (t*100)
1505 IF j<0 THEN LET aa=USR 40000: RETURN
1510 BEEP .01,j/4-20
1512 IF j>175 THEN GO TO 1520
1515 PLOT i,j: IF j<16 AND i/8>xc AND i/8<xc+2 THEN GO TO 1800
1517 IF (175-j)/8>xg AND (175-j)/8<xg+2 AND i/8>yg AND i/8<yg+2 AND dado=0 THEN GO TO 1900
1520 NEXT i: RETURN
1700 INK 0: PLOT 4,163: DRAW 56,0: DRAW 0,-16: DRAW -56,0: DRAW 0,16
1710 PAPER 7: PRINT AT 0,0;" TIEMPO ": FOR n=1 TO 3: PRINT AT n,0;"        ": NEXT n
1720 INK 0: PLOT 4,163: DRAW 56,0: DRAW 0,-16: DRAW -56,0
1730 DRAW 0,16: PAPER 5: RETURN
1800 LET aa=USR 40000: PRINT AT 20,xc; FLASH 1;"■■";AT 21,xc; FLASH 1;"■ "
1810 PAUSE 75: PRINT AT 20,xc;"  ";AT 21,xc;"  ": RETURN
1900 LET aa=USR 40000: PRINT AT xg,yg; FLASH 1;"■■";AT xg+1,yg; FLASH 1;"■ ": PAUSE 75
1910 PRINT AT xg,yg;"  ";AT xg+1,yg;"  ": LET dado=1: NEXT i
2000 INK 1: CLS : PRINT AT 2,7;"FUNDAMENTO TEORICO": INK 0
2010 PRINT AT 5,0;"-El movimiento parabolico es su-ma de dos movimientos:"
2020 PRINT AT 8,1;"-Uno horizontal, (segun X), con  velocidad cte y ecuacion:"
2030 PRINT AT 11,5;"X= V0.cos .t        (1)"
2040 PRINT AT 13,1;"-Y uno vertical (segun Y) uni- formemente decelerado:"
2050 PRINT AT 16,5;"Y= V0.sen .t-1/2 gt      (2)"
2060 PRINT AT 18,1;"-Dividiendo (2)/(1) obtenemos  la trayectoria:"
2070 PRINT AT 21,5;"Y=X[tg -gX/(2V0 cos  )]  (3)"
2080 PRINT #0;TAB 7;"(pulsa una tecla)": PAUSE 0: CLS
2090 PRINT AT 2,1;"-Para obtener el alcance del    disparo, hacemos Y=0 en (3) y   obtenemos X:"
2100 PRINT AT 6,5;"Xmax=V0 sen2 /g        (4)"
2110 PRINT AT 8,2;"(Ya que 2sen cos  = sen2 )"
2120 PRINT AT 10,1;"-La altura maxima se obtiene enel centro de la parabola. Haciendo en (3) X=Xmax/2 tenemos:"
2130 PRINT AT 14,5;"Ymax=V0 sen /(2g)        (5)"
2140 PRINT #0;" (pulsa una tecla para volver)": PAUSE 0: CLS : GO TO 75
3000 CLS : PRINT AT 0,8; INK 1;"EXPERIMENTACION";AT 3,1; INK 0;"-Aqui puedes practicar disparossobre un globo y una casa,intro-duciendo la velocidad inicial y el angulo inicial."
3010 PRINT AT 9,1;"-La situacion del globo y la casa se fija introduciendo:"
3020 PRINT AT 12,3;"-Las distancias que quieras";AT 14,3;"-A para situacion arbitraria";AT 16,3;"-R para repetir situacion";AT 18,3;"-P para la situacion propues-   ta en el problema"
3030 PRINT #0;TAB 7;"(pulsa una tecla)": PAUSE 0
3100 CLS : PRINT AT 2,5;"Vel.inicial: ";M(1,4);AT 4,5;"angulo    : ";M(2,4);AT 6,5;"alt. globo: ";M(3,4);AT 8,5;"dist. globo: ";M(4,4);AT 10,5;"dist. casa: ";M(5,4)
3110 PRINT AT 16,0;"INTRODUCE DATOS": LET a=12345: LET p=54321
3120 INPUT "Vel. inicial (m/s) V0= ";dato: LET cont=2: GO SUB 3500: LET v0=dato
3130 INPUT "Angulo inicial (grados)= ";dato: LET cont=4: GO SUB 3500: LET alfag=dato: LET alfa=alfag/180*PI
3140 INPUT "altura globo ((=340m)=) ";dato: LET cont=6: GO SUB 3500: LET ag=dato
3150 INPUT "distancia globo ((=450m)=) ";dato: LET cont=8: GO SUB 3500: LET dg=dato
3160 INPUT "distancia casa ((=450m)=) ";dato: LET cont=10: GO SUB 3500: LET dc=dato
3170 LET xc=INT (dc/500*32): LET yg=(dg/500*32): LET xg=(22-ag/500*32)
3200 GO SUB 1000: INK 0: PRINT AT 0,10;"O para otro disparo";AT 2,10;"E empieza de nuevo"
3205 PRINT #0;"Altura maxima ";INT hmax;"m. Alcance ";INT dis;"m"
3210 IF INKEY$="e" THEN CLS : GO TO 75
3220 IF INKEY$="o" THEN GO TO 3100
3230 GO TO 3210
3500 LET co=cont/2: IF dato=r THEN LET dato=M(co,4): GO TO 3540
3510 IF dato=a THEN LET dato=INT (RND*M(co,1)/2+M(co,2)+M(co,1)/4): GO TO 3540
3520 IF dato=p THEN LET dato=M(co,3): GO TO 3540
3530 IF dato<M(co,2) OR dato>M(co,2)+M(co,1) THEN GO TO 3110+cont*5
3540 LET M(co,4)=dato: PRINT AT cont,18;dato;"  ": RETURN
4000 CLS : DIM a$(11,25): PRINT AT 3,1;"PROBLEMA (unid: m, m/s, grados)"
4010 LET a$(2)="-angulo inicial": LET a$(1)="-Vel. inicial V0": LET a$(7)="-tiempo t de impacto"
4020 LET a$(3)="-altura del globo": LET a$(4)="-distancia hasta el globo": LET a$(6)="-volar el globo:"
4030 LET a$(5)="-distancia a la casa": LET a$(8)="-volar la casa:": LET a$(9)="Volar el globo y la casa:": LET a$(10)="(dar sus coordenadas)"
4040 PRINT AT 5,2;"-DATOS-";AT 14,2;"-SE PIDE-"
4050 PRINT #0;" (Ver primero si hay solucion)"
4060 DIM p(10,6): FOR i=1 TO 10: FOR j=1 TO 6
4070 READ a: LET p(i,j)=a: NEXT j: NEXT i
4080 DATA 2,5,11,8,1,7,1,5,11,8,2,7,2,3,4,6,1,7,1,3,4,6,2,7,3,4,5,9,1,2,2,1,11,8,10,11,2,1,3,6,4,7,2,1,4,6,3,7,2,1,7,6,10,11,7,9,11,16,18,20
4090 LET pro=INT (RND*9+1)
4100 FOR i=1 TO 6: PRINT AT p(10,i),3;a$(p(pro,i))
4110 IF i>3 OR p(pro,i)>7 OR p(pro,i)=6 THEN GO TO 4180
4120 LET let=p(pro,i): IF let=7 THEN PRINT AT p(10,i),29;INT (10+RND*15): GO TO 4180
4130 LET M(let,3)=INT (RND*M(let,1)/2+M(let,2)+M(let,1)/4): PRINT AT p(10,i),29;M(let,3)
4180 NEXT i
4185 PRINT AT 0,2;"(puedes pulsar A para ayuda)"
4190 IF p(pro,1)=3 THEN LET M(4,3)=M(5,3)-50: PRINT AT 9,29;M(4,3);"  "
4200 PAUSE 0: IF INKEY$="a" THEN GO SUB 4300
4220 RESTORE 4080: CLS : GO TO 75
4300 PRINT AT 0,1;"utiliza por orden las ec. ";
4305 IF pro=6 THEN PRINT 4;"  ": PAUSE 0: RETURN
4310 DIM q(9,2): FOR i=1 TO 9: FOR j=1 TO 2: READ qq: LET q(i,j)=qq: NEXT j: NEXT i
4320 DATA 4,1,4,1,3,1,3,1,3,4,4,0,3,1,3,1,1,2
4330 PRINT q(pro,1);" y ";q(pro,2): PAUSE 0: RETURN
```

«Shadow of the Unicorn». Microgen. Aventura Gráfica

# LA LEYENDA DEL UNICORNIO

La paz y la armonía reinaban entre las gentes que poblaban ambos lados del río Milfair. Durante siglos, desde que Ilis Clair —El Unicornio— expulsó a las fuerzas del mal de aquellas tierras y creó a todas las razas que actualmente allí habitaban: Hombres, Dwarfs, Amarines, Sandinids y Topils, habían vivido felices dedicados a sus trabajos cotidianos. Para ellos no existía el odio ni el rencor y jamás habían tenido que sufrir los horrores de una guerra.

Pero todo cambió el día en que Athron, un joven labrador, descubrió enterrado en sus campos un misterioso y envejecido libro. Al intentar abrirlo, el labrador se percató de que tenía una pequeña cerradura, por lo que le era totalmente imposible ver su interior. A pesar de su curiosidad tuvo que dejarlo olvidado en un cajón.

Al cabo de un tiempo, una diminuta llave de plata es entregada a la mujer del labrador por un misterioso Amarin que, casualmente, se enteró del descubrimiento del libro.

Al abrirlo, todas las fuerzas del mal surgen de improviso de entre sus páginas, introduciéndose en cada una de las personas que allí habitaban y convirtiéndolas en seres malvados y ansiosos de destrucción.

Así fue como dos reinos se vieron súbitamente avocados hacia el caos y la perdición.

## La misión

Consiste en capturar al diablo que ha desencadenado todas estas catástrofes y encerrarlo de nuevo en el libro del cuál surgió, poniendo así fin a este periodo de conflictos y muertes inútiles y devolviendo a todas las razas la paz que desde siempre habían disfrutado.

Para ello, nos ayudaremos de los 10 diferentes personajes que configuran la historia y de una serie de objetos que éstos irán encontrando durante sus largos y tortuosos viajes.

El uso adecuado de cada uno de ellos nos irá abriendo nuevos caminos y poco a poco, con muchísima paciencia y perseverancia, conseguiremos avanzar en nuestra complicada misión.

## Los gráficos

Los gráficos no son del tipo multicolor a que nos tiene acostumbrados Microgen. En esta ocasión han







## LOS PERSONAJES

**1 MITHULIN** (Heredero de Oronfal).

Oronfal es el reino al Este del río Milfair. Mithulin es el hijo de Theltien, rey de Oronfal, que con su gran ejército derrota a los hombres del Oeste en Oslar.

**2 AVARATH** (Mago de Zim-Farinid).

Mago de grandes poderes, que junto con Holdin combate contra los Nalesh (bestias diabólicas) y se dirige a Clarooth.

**3 HOLDIN** (Capitán de Falforn).

Falforn es el reino al Oeste del Milfair. Es el Capitán de las tropas que, al abrirse el libro, se matan entre sí. Escapa al bosque donde se encuentra, en las ruinas de Glowist, con Avarath.

**4 ROLQUIN** (Reina de Falforn)

Bella dama que gusta de mezclarse con sus subditos. A través de ellos descubre los desastres que están teniendo lugar.

**5 SHARMER** (Capitán de los Dwarf).

Los Dwarf son una raza que habita en Rimersel. El fue quien comenzó la guerra enfrentándose primero a sus propios compatriotas para después aliarse a las tropas de Lairmath, las cuales fueron derrotadas en Oslar por Mithulin.

**6 ULIN-GAIL** (Sátiro de Harvena)

Fauno de la raza Amarin, colaborador de Mithulin.

**7 GUINOL** (Miembro de la raza de los Topil)

Ayuda a Avarath y a Holdin en su difícil periplo.

**8 VILYAN** (Miembro de la raza de los Topil)

Advierte a Theltiem que la guerra que va a tener lugar es un grave error y su pueblo no participará en ella.

**9 KIELMATH** (Maga en Zim-Farinid)

Es la poseedora actual del libro maldito que desencadena todas las tragedias.

**0 LAIRMATH** (Capitán de Falforn)

Luchó contra los Dwarf, se alió posteriormente a ellos y finalmente, fue derrotado por Mithulin.

preferido personajes monocromos desplazándose por un fondo que consta principalmente de franjas bicolores, lo que simplifica enormemente el sistema. De esta forma, pueden hacerse un mayor número de pantallas.

Otra particularidad que los hace especialmente atractivos es que se ha conseguido imprimir un efecto de tridimensionalidad al paisaje haciendo que el tamaño de los edificios dependa de la distancia a que nos encontramos de ellos.

### El movimiento

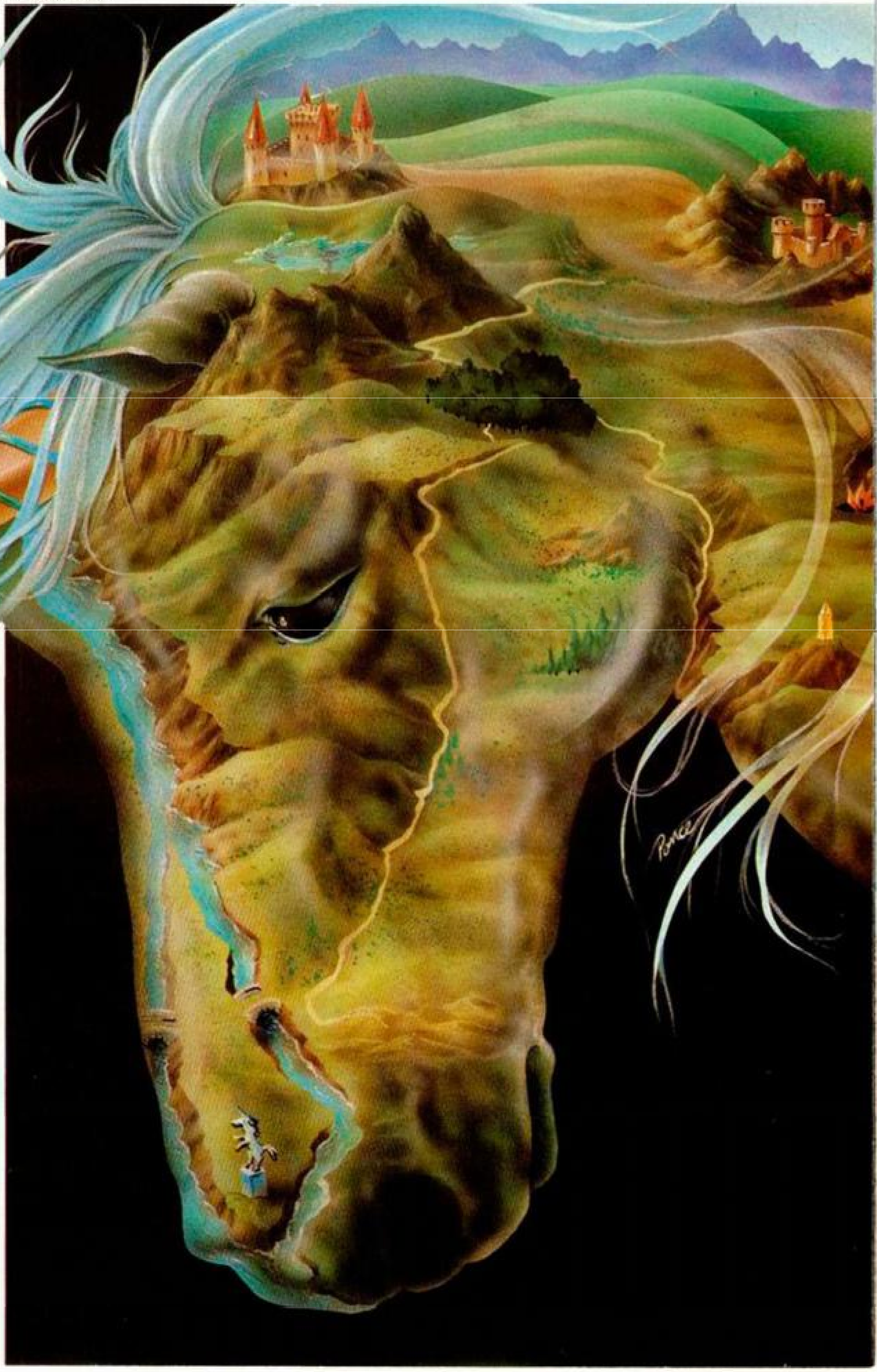
Para conseguir que los personajes tengan la posibilidad de moverse libremente en cualquier sentido y que al mismo tiempo sepamos en cada momento dónde están y qué dirección llevan, los hábiles programadores de Micro-Gen han desarrollado un ingenioso sistema de movimiento.

Los personajes sólo pueden desplazarse hacia la izquierda o hacia la derecha de la pantalla, pero con el simple hecho de pulsar una tecla podemos hacer que éstos giren sobre sí mismos, de tal manera que cambie su orientación y pueden tomar la dirección deseada.

Gracias a una pequeña brújula situada en el extremo superior de la pantalla podemos conocer en todo momento el plano de nuestra visión.

### El Microplus

Se trata de una experiencia inédita en juegos para Spectrum. Por primera vez se ha conseguido que un programa tenga 64K de memoria real gracias al nuevo periférico «Microplus». En esencia se trata de una memoria ROM suplementaria que alberga los primeros 16K del juego. Esta memoria






**«La llave de plata giró lentamente entre los dedos de la joven y el libro cayó abierto por su primera página...»**

# ¡NUEVO!

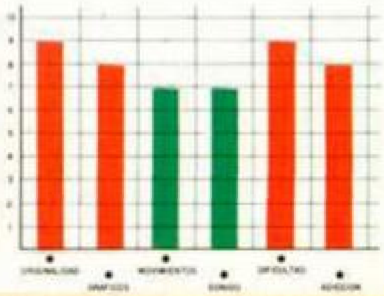
**«En este libro he puesto las cosas que el hombre y la bestia deben exigir para protegerlas del ojo inquisidor y arrebatarlas de la mano que acapara. Pero en este libro hay una trampa...»**

pagina sobre la ROM del Spectrum, inhabilitándola. De esta forma se consiguen una cantidad extra de gráficos, caracteres redefinidos, etc. En este juego hay más de 6000 pantallas distintas, aunque algunas son bastante parecidas entre sí.

*OSLAR*

**En esta isla se encuentra la magnífica estatua de Ilis Clair, el Unicornio. Escenario de la batalla entre las tropas del Este y del Oeste.**



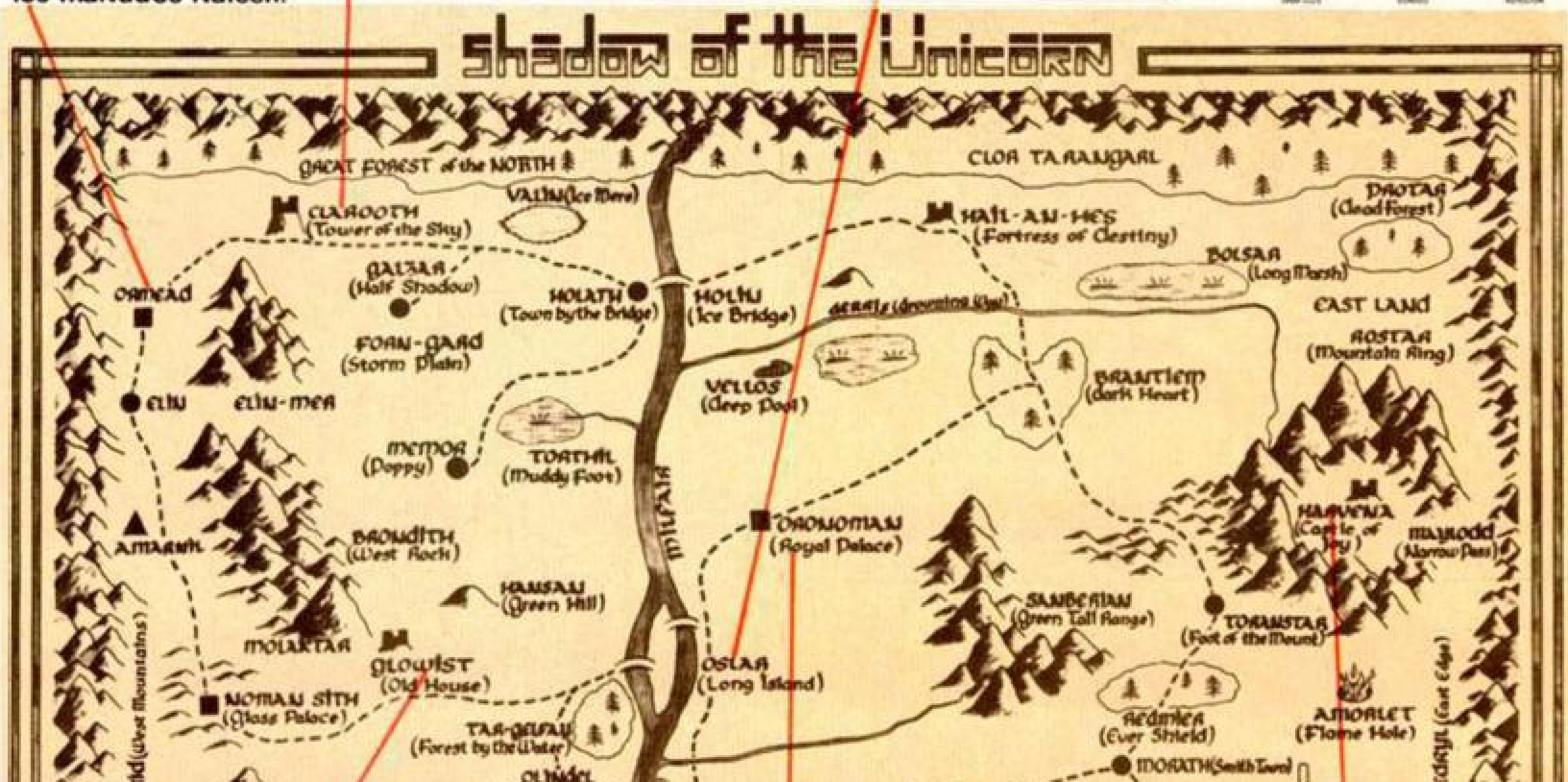



*CLAROOTH*

**Capital de Falforn. En ella se encuentra la Torre del Cielo, donde habita la Reina Rolquin.**

*ORMEAD*

**A su paso por esta ciudad, Avarath y Holdin, descubren que ha sido destruida por los malvados Nalesh.**







*GLOWIST*

**Ruinas de un antiguo palacio destruido, escenario donde se encuentran Avarath y Holdin.**

*RIMERSEL*

**Ciudad situada al sur de Falforn, donde viven los Dwarf.**

*ORONOMAN*

**Capital del Reino de Oronfal, donde viven Theltiem y su heredero Mithulin.**

*MURE*

**Aglomeración de dunas al sur de Oronfal donde Mithulin mantiene una importante conversación con su malherido padre.**

*HARVENA*

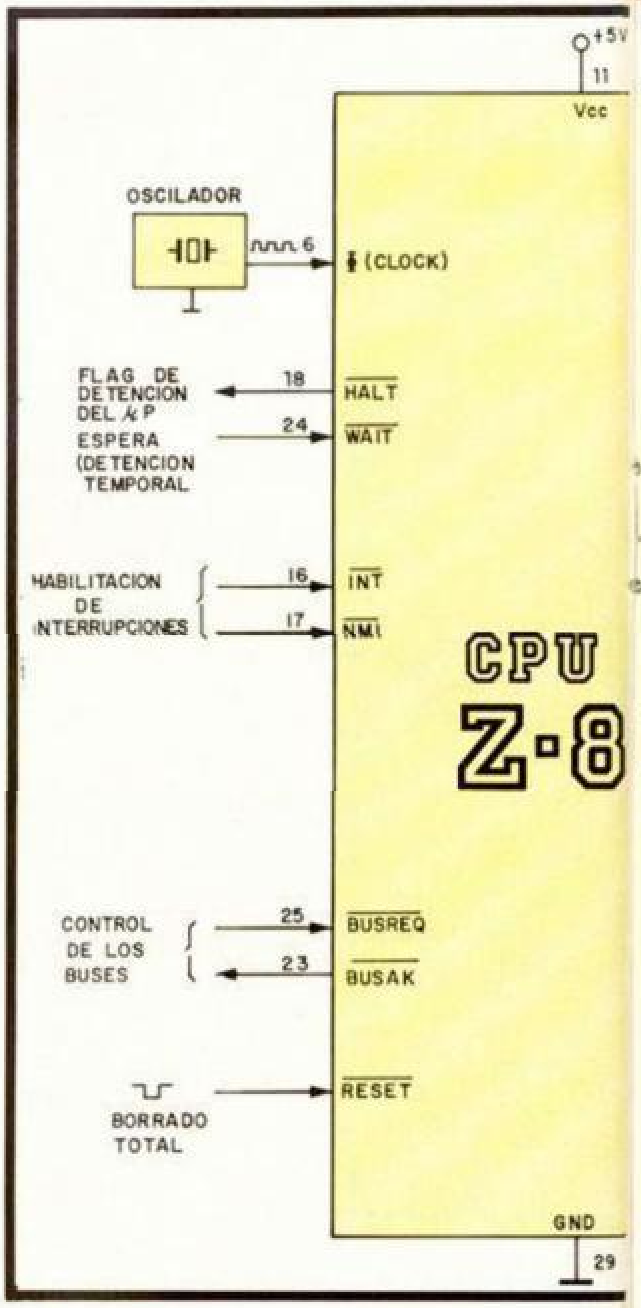
**Ciudad milenaria en el extremo oriental de Oronfal. Maylodd es la única entrada posible. En ella habitan los Amarin.**


WEST BANK
Defiende el
banco de Soft City
del ataque de los forajidos
y consigue escribir tu nombre
en la leyenda del «FAR WEST».
SPECTRUM 48 K y PLUS, 1.950 ptas.
Fotos tomadas de un Spectrum 48 K
OPERACION PUZZLE
Dinamic Software pone
en marcha la operación
puzzle: regalarte
6 millones de pesetas
en programas.
* Más información en los originales
DINAMIC
«MANSION DINAMIC»
c/ Tilos, 2, 21, MONTEPRINCIPE
Boadilla del Monte. MADRID
TEL.: 715 00 67
TIENDAS: 447 34 10

# EL MICROPROCESADOR Z-80 (II)

Primitivo de FRANCISCO

**El microprocesador Z-80 cobija bajo su negro caparazón, en la diminuta pastilla de silicio, un buen montón de conceptos y filosofías funcionales, fruto de años de inventiva y evolución en el mundo de los microcircuitos y del Software. En esta segunda parte proseguimos con su descripción para facilitar la comprensión del hardware que lo rodea.**




**Descripción de los terminales del microprocesador Z-80.**

El Z-80 se comunica con el exterior por 40 terminales los cuales se reparten la totalidad de las funciones operativas del microprocesador. Todos ellos funcionan con los niveles lógicos TTL porque es ésta la familia de circuitos integrados más divulgada, la cual, además ha conseguido el mayor alto grado de perfección, tanto en velocidad de conmutación como en consumo eléctrico a precios económicos. Los niveles lógicos de tensión en TTL son: de ØV hasta Ø'8V para el cero lógico y desde 2,4 hasta 5V para el uno lógico.

Las salidas del Z-80 acceden hacia el exterior a través de bufferes que amplifican y proporcionan a las mismas la potencia para dar señal a varias entradas TTL que se pueden conectar en paralelo. El número de éstas depende de la variante TTL que se le conecte. Frecuentemente se usa la TTL-LS, la más avanzada, que posee un menor FAN-In o corriente de entrada. Como se puede observar en todos nuestros montajes es ésta la que utilizamos, perfectamente reconocible por las siglas LS que se encuentran en medio del código comercial, por ejemplo: el circuito 74240 de TTL se convierte en 74LS240 en TTL-LS.

Como características eléctricas del Z-80 hay que añadir que es capaz de disipar hasta 1,5 watios, que soporta unas temperaturas de almacenamiento entre –65° C y 150° C y que todas sus entradas admiten un margen de tensión desde –0,3V hasta 7V respecto a tierra (entre los cuales se encuentran, naturalmente, los niveles mencionados TTL). La descripción funcional de cada uno de sus terminales es: (Vease figura uno).

## AØ - A15

Es el bus de direcciones, que sale hacia el exterior mediante 16 terminales.

## DØ - D7

Es el bus de datos, se comunica con el exterior mediante 8 terminales, el buffer interior es triestado para permitir el trasiego de información tanto hacia el Z-80 como desde el Z-80.

## WR

Sale por el pin 22. Cuando el Z-80 pone este terminal a cero lógico o nivel bajo indica a los periféricos o a la memoria que en este momento se va a producir una operación de escritura (Write). Esto implica que el bus de datos se convierta en bus de salida para entregar por sus ocho bits el byte adecuado.

## RD

Sale por pin 21. Cuando el Z-80 lo pone a nivel bajo quiere indicar a la memoria y periféricos que en este momento se produce un ciclo de lectura (Read) y el bus de datos se pone en condiciones de recibir un byte. (Nunca se podrán dar RD y WR al mismo tiempo).

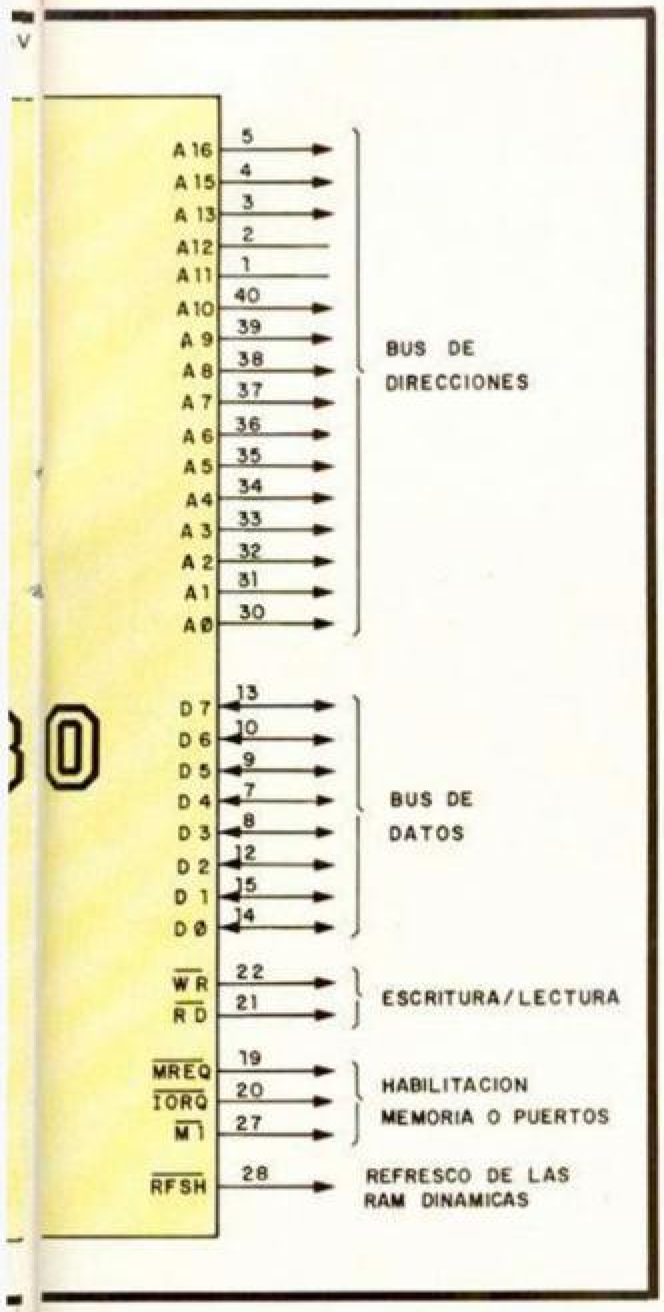
## MREQ

Sale por el pin 19. El Z-80 pone este terminal a nivel bajo para indicar que la operación que se inicia se refiere a la memoria, a la cual tiene que acceder para leer o escribir (MREQ = MEMORY REQUEST).

Esta señal se producirá lógicamente en combinación conlas señales RD o WR y con el bus de direcciones y de datos para acceder a una de las 65536 direcciones posibles de memoria (RAM o ROM).

## IORQ

Sale por el pin 20. El microprocesador pone a cero este terminal para indicar a los periféricos que ha de acceder a alguno de ellos para leer o escribir según la dirección que presenta al tiempo por el bus de direcciones con el dato a transferir por el bus de datos. (IORQ= INPUT/OUTPUT REQUEST). El Z-80 posee también 65536 direcciones de periféricos llamadas puertos («ports»), sólo que el acceso a los puertos altos es




algo más complicado que el de memoria, requiere la ejecución de·dos instrucciones consecutivas: IORQ y MREQ. Son antagónicas y nunca se producirán al tiempo.

## M1

Sale por el pin 27. Este terminal pasa a nivel bajo mientras se esté procesando el byte de código de toda instrucción. M1 también pasa a nivel bajo con la señal IORQ durante el ciclo de reconocimiento de alguna interrupción. (M1 = MACHINE CYCLE ONE).

## RFSH

Sale por el pin 28. Este terminal pasa a nivel bajo para que en el ciclo de refresco de las memorias RAM dinámicas, que se conecten al Z-80, la dirección de refresco salga hacia el exterior por los siete bits de dirección (AØ a A6). Esta señal se produce en combinación con MERQ que hace de clock de refresco. A7 permanece a cero y A8 a A15 presentan el contenido del registro interno I. El proceso de refrescos es complejo y de él trataremos más adelante.

## Vcc

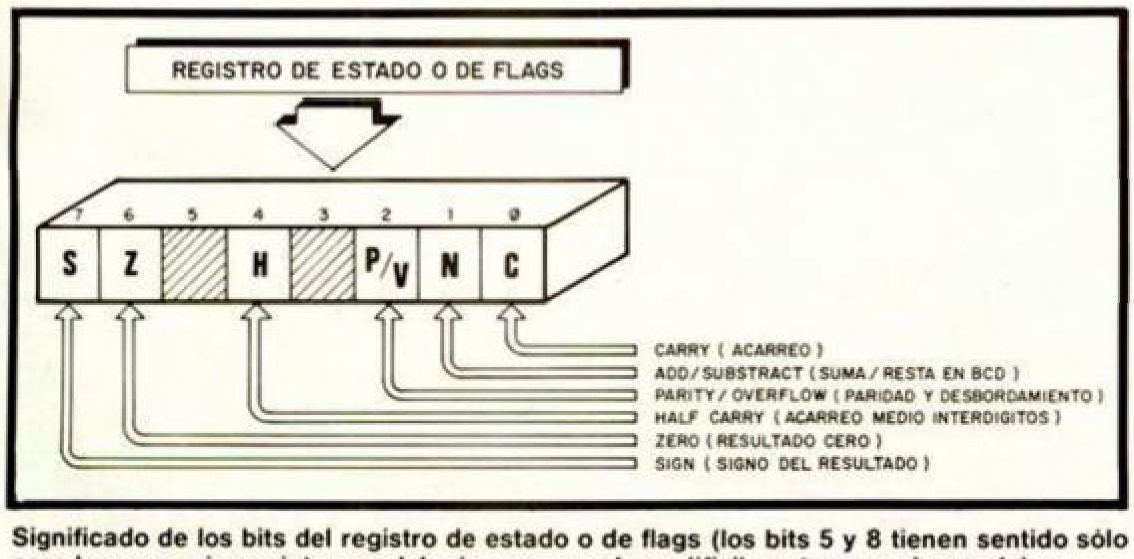
Entrada de alimentación (+ 5V), pin 11. Sobre esta entrada el Z-80 presenta el consumo de corriente hacia la fuente de 5V, esta corriente es próxima a 200 mA.

## GND

Es la conexión a tierra (Ground). Sale por el terminal 29.

## CLOCK

Esta senãl también se suele representar por la letra griega $\Phi$ para indicar entrada de reloj de una única fase. Se corresponde con el pin 6 y por ella entra la señal de reloj (que en el Spectrum es de 3,5 Mhz), que le proporciona la ULA (500 Mhz menos de los que podría soportar el Z-80 A que es el que equipa al Spectrum).




**Significado de los bits del registro de estado o de flags (los bits 5 y 8 tienen sentido sólo para las operaciones internas del microprocesador y difícilmente para el usuario).**

## HALT

Señal de salida por el pin 18; cuando el Z-80 la pone a nivel bajo indica al exterior que se ha producido una instrucción HALT y que, hasta que no sea sacado de este estado mediante un reset o una interrupción, el Z-80 mantendrá el refresco de memoria ejecutando constantemente instrucciones NOP (no operación).

## WAIT

Entra por el pin 24; cuando algún dispositivo exterior lleva a este terminal a nivel bajo, el Z-80 permanecerá inactivo mientras dure la señal WALT, manteniendo intactos los registros y el contenido de los buses, pero cesando el refresco de las memorias dinámicas. Es por esto que en el Spectrum esta señal no debe durar más de 2 milisegundos. Se puede usar para sincronizar al Z-80 con periféricos lentos que tienen un tiempo de conmutación superior al del Z-80. (Wait = Espera).

## INT

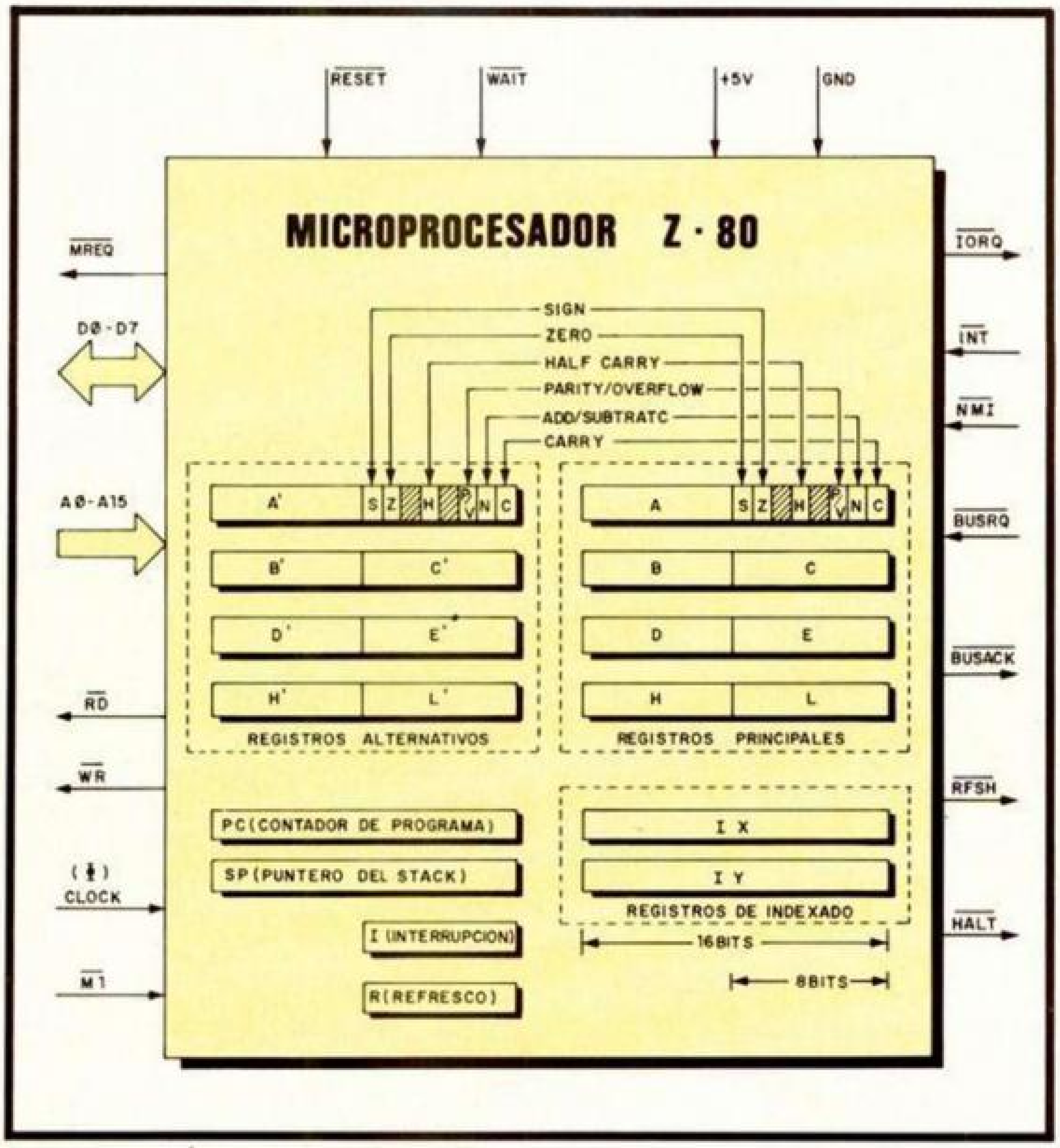
Pin 16; es la entrada de las interrupciones enmascarables.

## NMI

Entra por el pin 17 y habilita las interrupciones no enmascarables. (NMI = NOM MASKABLE INTERRUPT). De las señales INT y NMI trataremos profundamente en un próximo apartado hardware.

## BUSREQ

Entra por el pin 25. Es puesta a nivel bajo por algún periférico que tiene que compartir los buses del Z-80. Cuando un periférico hace petición de acceso a los buses, el Z-80 pasa a atenderlo tan pronto como concluya el ciclo máquina en ejecución e inmediatemente pone sus buses en alta impedancia para




**Registros del microprocesador Z-80 y funciones de sus conexiones con el exterior.**

permitir entrar en ellos a otro periférico o microprocesador. (BUSREQ = BUS REQUEST).

## BUSACK

Es una salida por donde el Z-80 le comunica al periférico que solicitó por Busreq acceso a los buses, que su petición pasa a ser atendida. Para ello pone esta salida a nivel bajo. Y a alta impedancia sus buses, los cuales llevan, como ya se ha dicho, circuitos amplificadores de salida tipo triestado.

Mientras que el Z-80 se encuentra en aceptación de buses compartidos abandona el refresco de las memorias RAM DINAMICAS que se encuentran colgadas de él. Esta condición ha de estar prevista necesariamente en el desarrollo hardware que use esta opción. (Busak = bus acknowlege)

Esta salida se efectúa con el pin 25 del microprocesador.

## RESET

Es una entrada al Z-80 por el pin 26. Mediante ella se consigue inicializar en todo momento al microprocesador, llevando esta entrada a nivel lógico bajo. Cuando esto ocurre todos los registros internos son puestos a cero, incluso el registro contador de programa. Al llevar a uno este terminal inmediatamente el Z-80 comienza a ejecutar lo que se halle a partir de la dirección ØØØØ H.

Durante el reset se producen también algunas otras condiciones que hay que conocer:

— Habilitación de las interrupciones.

— El registro de refresco R se pone también a cero iniciándose a partir de este momento un nuevo ciclo.

— El registro I que vectoriza un modo de interrupciones también se pone a cero.

— Con el reset se selecciona automáticamente el modo Ø de interrupción (IMØ).

Mientras que esta entrada está a nivel bajo, todos los buses se ponen a alta impedancia y cesa el refresco de las memorias RAM. Al mismo tiempo, el resto de las señales se ponen en estado inactivo. El reset funciona por nivel, no por flanco, por lo cual este estado puede prolongarse indefinidamente siempre que el reset esté a nivel lógico cero.

## Registros del Z-80

Un registro no es más que una cadena de biestables tipo D que pueden almacenar en paralelo una iformación binaria temporalmente. Su uso se hace imprescindible para toda la operatividad interna del microprocesador, para la comunicación con el exterior y para el almacenamiento temporal que sirve de gran auxilio al programador.

Ya mencionamos brevemente los nombres de la totalidad de los registros; ahora los analizaremos detenidamente.

Existen dos bancos de registros idénticos compuestos ambos de cuatro registros de 16 bits. Estos dos bancos reciben el nombre de registros principales y registros alternativos.

Cada uno de estos registros pueden ser usados como registros de ocho bits o de dieciseis; para ello basta con usar las instrucciones apropiadas que los tratan como simples o como dobles. Así por ejemplo, el registro BC puede ser usado como B, como C o como BC, esta última configuración posibilita alcanzar directamente la totalidad del espacio direccionable (64 K) y operar con valores de hasta 65535.

Ambos bancos de registros se corresponden mutuamente, el BC con el B'C', el DE con el D'E' y el HL con H'L'. Todos estos pueden ser usados indistintamente por el programador excepto en algunas instrucciones que requieren registros específicos.

Los registros AF y A'F' son un tanto especiales y de ello habremos de tratar más extensamente en su momento.

El registro A es el registro acumulador por el que pasan todos los resultados operativos de ocho bits y a él están referidas la mayoría de las instrucciones. Es, por tanto, un registro muy específico. El registro F o de FLAGS es el que lleva el estado del microprocesador en todo momento. Cada uno de sus ocho bits tiene un cometido, incluso los dos que en la figura dos aparecen rayados.

| AND | Ø | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 1 | Ø | 1 | Ø | 1 | Ø | 1 | Ø | 1 | Ø | 1 | Ø | 1 | Ø | 1 | Ø | 1 |
| 2 | Ø | Ø | 2 | 2 | Ø | Ø | 2 | 2 | Ø | Ø | 2 | 2 | Ø | Ø | 2 | 2 |
| 3 | Ø | 1 | 2 | 3 | Ø | 1 | 2 | 3 | Ø | 1 | 2 | 3 | Ø | 1 | 2 | 3 |
| 4 | Ø | Ø | Ø | Ø | 4 | 4 | 4 | 4 | Ø | Ø | Ø | Ø | 4 | 4 | 4 | 4 |
| 5 | Ø | 1 | Ø | 1 | 4 | 5 | 4 | 5 | Ø | 1 | Ø | 1 | 4 | 5 | 4 | 5 |
| 6 | Ø | Ø | 2 | 2 | 4 | 4 | 6 | 6 | Ø | Ø | 2 | 2 | 4 | 4 | 6 | 6 |
| 7 | Ø | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Ø | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| 9 | Ø | 1 | Ø | 1 | Ø | 1 | Ø | 1 | 8 | 9 | 8 | 9 | 8 | 9 | 8 | 9 |
| A | Ø | Ø | 2 | 2 | Ø | Ø | 2 | 2 | 8 | 8 | A | A | 8 | 8 | A | A |
| B | Ø | 1 | 2 | 3 | Ø | 1 | 2 | 3 | 8 | 9 | A | B | 8 | 9 | A | B |
| C | Ø | Ø | Ø | Ø | 4 | 4 | 4 | 4 | 8 | 8 | 8 | 8 | C | C | C | C |
| D | Ø | 1 | Ø | 1 | 4 | 5 | 4 | 5 | 8 | 9 | 8 | 9 | C | D | C | D |
| E | Ø | Ø | 2 | 2 | 4 | 4 | 6 | 6 | 8 | 8 | A | A | C | C | E | E |
| F | Ø | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

**Fig. 5. Tabla hexadecimal de "AND".**

| XOR | Ø | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ø | Ø | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 1 | 1 | Ø | 3 | 2 | 5 | 4 | 7 | 6 | 9 | 8 | B | A | D | C | F | E |
| 2 | 2 | 3 | Ø | 1 | 6 | 7 | 4 | 5 | A | B | 8 | 9 | E | F | C | D |
| 3 | 3 | 2 | 1 | Ø | 7 | 6 | 5 | 4 | B | A | 9 | 8 | F | E | D | C |
| 4 | 4 | 5 | 6 | 7 | Ø | 1 | 2 | 3 | C | D | E | F | 8 | 9 | 4 | B |
| 5 | 5 | 4 | 7 | 6 | 1 | Ø | 3 | 2 | D | C | F | E | 9 | 8 | B | A |
| 6 | 6 | 7 | 4 | 5 | 2 | 3 | Ø | 1 | E | F | C | D | A | B | 8 | 9 |
| 7 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Ø | F | E | D | C | B | A | 9 | 8 |
| 8 | 8 | 9 | A | B | C | D | E | F | Ø | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 9 | 9 | 8 | B | A | D | C | F | E | 1 | Ø | 3 | 2 | 5 | 4 | 7 | 6 |
| A | A | B | 8 | 9 | E | F | C | D | 2 | 3 | Ø | 1 | 6 | 7 | 4 | 5 |
| B | B | A | 9 | 8 | F | E | D | C | 3 | 2 | 1 | Ø | 7 | 6 | 5 | 4 |
| C | C | D | E | F | 8 | 9 | A | B | 4 | 5 | 6 | 7 | Ø | 1 | 2 | 3 |
| D | D | C | F | E | 9 | 8 | B | A | 5 | 4 | 7 | 6 | 1 | Ø | 3 | 2 |
| E | E | F | C | D | A | B | 8 | 9 | 6 | 7 | 4 | 5 | 2 | 3 | Ø | 1 |
| F | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Ø |

**Figura 6. Tabla hexadecimal de "EXOR".**

una operación "EXOR" de un número consigo mismo, el resultado es siempre "cero"; lo cual es muy útil, ya que si se quiere cargar el número "cero" en el acumulador, se puede utilizar la instrucción "XOR A" ("EXOR" del acumulador consigo mismo) que ocupa la mitad de memoria y es el doble de rápida que "LD A,Ø" (cargar el acumulador con cero).

De la misma forma, si se hace un "OR" o un "AND" de un número consigo mismo, éste no varía. En el programa de ejemplo de este capítulo, hemos usado la instrucción "AND A" ("AND" del acumulador consigo mismo) para poner a cero el indicador de acarreo sin que varíe el contenido del acumulador.

El empleo de operadores lógicos nos va a permitir movernos por tablas, calcular direcciones de memoria, poner "máscaras" a algunos bits, y un sinfín de utilidades que justifican la necesidad de adquirir el mayor dominio posible del álgebra de Boole.

## EJERCICIOS

1.- Realizar un "AND", un "OR" y un "EXOR" entre los siguientes pares de números:

```
1 1 1 Ø Ø 1 Ø Ø     y     Ø 1 1 Ø 1 Ø Ø 1
Ø 1 Ø 1 Ø Ø Ø Ø     y     1 Ø 1 Ø 1 1 1 1
Ø Ø Ø 1 1 Ø 1 Ø     y     1 Ø 1 1 Ø 1 Ø Ø
      2Ah           y          BFh
      B1h           y          58h
```

2.- Repetir el ejercicio anterior, utilizando los complementarios de los números propuestos.





# INTRODUCCION AL CODIGO MAQUINA

Prácticamente cualquier usuario de Spectrum ha tenido alguna vez contacto con el código máquina. En este lenguaje están escritos los mejores programas comerciales, y algunas veces lo hemos utilizado en las páginas de nuestra revista.

Los menos experimentados se preguntan qué es eso del código máquina, qué tiene que ver con sentencias como DATA, USR, RANDOMIZE, etc. y sobre todo, para qué sirve. A lo largo de este curso, vamos a dar respuesta a estas y otras preguntas.

El código máquina no es sólo otro lenguaje más de programación; se trata de hablarle al ordenador directamente en el lenguaje que él entiende. De esta forma, no estamos sujetos a las restricciones del Basic, que son muchas, y tenemos un dominio completo sobre nuestra máquina.

Normalmente, nadie programa directamente en código máquina, este lenguaje está compuesto únicamente por sucesiones de números y no existe nadie capaz de recordarlos todos. Por esto, se suelen escribir los programas en Assembler y después, traducirlos a código máquina. Esta última tarea, se conoce por el nombre de "ensamblado", y habitualmente, se realiza con la ayuda de un programa llamado "Ensamblador".

En los cuatro primeros capítulos del curso, se estudian algunas nociones previas que serán necesarias en los capítulos posteriores, por lo que no es recomendable pasar a estudiar un capítulo sin haber comprendido totalmente el anterior.

Es muy probable que el



gran volumen de lectores nos haga imposible mantener una correspondencia personalizada, pero aún así, nos agradaría que quienes sigan el curso nos escriban contándonos sus progresos o las dificultades que encuentran. Estas cartas nos permitirán ir adaptando las explicaciones a un nivel que satisfaga a todos y permita que nadie se quede "descolgado".

En este mismo capítulo incluimos dos rutinas de utilidad escritas en código máquina que permiten hacer "Scroll" lateral de la pantalla, a derecha e izquierda y pixel a pixel.

Las dos rutinas se han ensamblado una a continuación de la otra y son reubicables, es decir, se pueden colocar en cualquier parte de la memoria. Nosotros las hemos ensamblado a partir de la dirección 55ØØØ, pero quien disponga sólo de 16 K, puede colocarlas en otra dirección, haciendo unas pequeñas modificaciones en el programa cargador, que explicaremos un poco más adelante.

En la FIGURA 1, reproducimos fotográficamente el listado en Assembler de las dos rutinas. No se preocupe el lector si le suena a "chino", un listado en Assembler no es más difícil de entender que uno en Basic, cuando se conoce. Al final del curso, más de uno será capaz de mejorarlo.

El PROGRAMA 1 sirve, lógicamente, para cargar estas rutinas en memoria sin necesidad de Ensamblador. De esta forma, no es necesario saber código máquina para usarlas. Una vez estén en memoria, basta teclear:

```
RANDOMIZE USR 55ØØØ
```

```
*HISOFT GENS3M ASSEMBLER*
        ZX SPECTRUM

 Copyright HISOFT 1983
 CURSO C/M MICROHOBBY

Pass 1 errors: 00

         10 *C-
         20 *D+
55000    30          ORG     55000
         40 ;
         50 ;SCROLL_IZQUIERDA
         60 ;
55000    70          LD      HL,22527
55003    80          LD      C,192
55005    90 BUC_2    LD      B,32
55007   100          AND     A
55008   110 BUC_1    RL      (HL)
55010   120          DEC     HL
55011   130          DJNZ    BUC_1
55013   140          JR      NC,NOCA_1
55015   150          LD      (VAR),HL
55018   160          LD      IX,(VAR)
55022   170          SET     0,(IX+32)
55026   180 NOCA_1   DEC     C
55027   190          JR      NZ,BUC_2
55029   200          RET
        210 ;
        220 ;SCROLL_DERECHA
        230 ;
55030   240          LD      HL,16384
55033   250          LD      C,192
55035   260 BUC_4    LD      B,32
55037   270          AND     A
55038   280 BUC_3    RR      (HL)
55040   290          INC     HL
55041   300          DJNZ    BUC_3
55043   310          JR      NC,NOCA_2
55045   320          LD      (VAR),HL
55048   330          LD      IX,(VAR)
55052   340          SET     7,(IX-32)
55056   350 NOCA_2   DEC     C
55057   360          JR      NZ,BUC_4
55059   370          RET
23728   380 VAR      EQU     23728

Pass 2 errors: 00

Table used:    97  from    160
```

**Fig. 1. Listado Assembler de las rutinas de "Scroll" lateral.**



OPERADOR AND

En cierto sentido, se puede considerar que este operador es el opuesto del anterior. Equivale al producto, en cuanto a las propiedades algebraicas que satisface.

Cuando operamos dos elementos de nuestro conjunto ("unos" o "ceros") obtenemos un "1" solamente si ambos elementos son "1"; y un "Ø" en cualquier otro caso. La Tabla de verdad del operador AND es la siguiente:

```
Ø AND Ø = Ø
Ø AND 1 = Ø
1 AND Ø = Ø
1 AND 1 = 1
```

Vamos a ver que ocurre si, con los números del ejemplo anterior, aplicamos la operación AND:

```
  Ø1ØØØ11Ø   (46h)
     AND
  111ØØØ11   (E3h)
      =
  Ø1ØØØØ1Ø   (42h)
```

Esta vez, hemos puesto un "1" sólo en los lugares donde ambos números tenían un "1", y hemos puesto "Ø" en todos los demás lugares. La tabla del operador "AND" en Hexadecimal, está representada en la FIGURA 5.

OPERADOR EXOR

No se trata propiamente, de una operación del álgebra de Boole, pero es necesario describirlo dado el gran uso que se hace de él cuando se programa en Assembler. El operador "EXOR" es, en cierta forma, una mezcla de los operadores "AND" y "OR".

Cuando operamos dos elementos de nuestro conjunto mediante este operador, obtenemos un "1", sólo si uno de los dos elementos es "1"; y obtenemos un "Ø" tanto si ambos son "ceros", como si ambos son "unos". La Tabla de verdad del operador "EXOR" es la siguiente:

```
Ø EXOR Ø = Ø
Ø EXOR 1 = 1
1 EXOR Ø = 1
1 EXOR 1 = Ø
```

Apliquemos este operador a los números del ejemplo anterior:

| NOT | Ø | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Ø |

**Fig. 3. Tabla hexadecimal de "NOT".**

| OR | Ø | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ø | Ø | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 1 | 1 | 1 | 3 | 3 | 5 | 5 | 7 | 7 | 9 | 9 | B | B | D | D | F | F |
| 2 | 2 | 3 | 2 | 3 | 6 | 8 | 6 | 7 | A | B | A | B | E | F | E | F |
| 3 | 3 | 3 | 3 | 3 | 7 | 7 | 7 | 7 | B | B | B | B | F | F | F | F |
| 4 | 4 | 5 | 6 | 7 | 4 | 5 | 6 | 7 | C | D | E | F | C | D | E | F |
| 5 | 5 | 5 | 8 | 7 | 5 | 5 | 7 | 7 | D | D | F | F | D | D | F | F |
| 6 | 6 | 7 | 6 | 7 | 6 | 7 | 6 | 7 | E | F | E | F | E | F | E | F |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | F | F | F | F | F | F | F | F |
| 8 | 8 | 9 | A | B | C | D | E | F | 8 | 9 | A | B | C | D | E | F |
| 9 | 9 | 9 | B | B | D | D | F | F | 9 | 9 | B | B | D | D | F | F |
| A | A | B | A | B | E | F | E | F | A | B | A | B | E | F | E | F |
| B | B | B | B | B | F | F | F | F | B | B | B | B | F | F | F | F |
| C | C | D | E | F | C | D | E | F | C | D | E | F | C | D | E | F |
| D | D | D | F | F | D | D | F | F | D | D | F | F | D | D | F | F |
| E | E | F | E | F | E | F | E | F | E | F | E | F | E | F | E | F |
| F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |

**Fig. 4. Tabla hexadecimal de "OR".**

```
  Ø1ØØØ11Ø   (46h)
    EXOR
  111ØØØ11   (E3h)
      =
  1Ø1ØØ1Ø1   (42h)
```

Hemos puesto un "Ø" en los lugares donde ambos números eran iguales (dos "unos" o dos "ceros"), y un "1" donde eran distintos ("cero" y "uno" o "uno" y "cero").

La tabla del operador "EXOR" en hexadecimal, está representada en la FIGURA 6. Como curiosidad importante, cabe señalar que si se realiza

y el producto en el álgebra a la que estamos acostumbrados. El hecho de que las operaciones sean internas, quiere decir que al operar dos elementos del conjunto, lo que se obtiene es otro elemento que también pertenece al mismo. Lo que tiene por consecuencia, que siempre que hagamos operaciones lógicas entre "ceros" y "unos", obtendremos indefectiblemente, "ceros" y "unos".

Puesto que un circuito electrónico (y los ordenadores lo son) no pueden trabajar más que con "ceros" y "unos", el álgebra de Boole parece un instrumento especialmente adecuado a la Informática. Como se explica en el capítulo que trata de los sistemas de numeración, agrupamos los "unos" y "ceros" en secuencias de 8 o 16 para componer otros números; pero en definitiva, estaremos trabajando con "ceros" y "unos", y el juego de instrucciones del microprocesador nos permite aplicar operaciones lógicas entre el contenido de los registros.

A continuación, vamos a ver uno a uno los operadores lógicos de nuestro álgebra de Boole.

OPERADOR NOT

No se trata propiamente de un operador lógico pero podemos considerarlo como tal. El operador NOT se aplica sobre un solo elemento del conjunto, y lo convierte en su complementario. Es decir, si aplicamos NOT sobre un "Ø", obtenemos un "1"; y viceversa, si aplicamos NOT sobre un "1", obtenemos un "Ø". Su "Tabla de verdad" sería la siguiente:

```
NOT Ø = 1
NOT 1 = Ø
```

Una "tabla de verdad" equivale, en álgebra de Boole, a la tabla de sumar o multiplicar. Se trata de una representación de todas las soluciones posibles que se pueden obtener con un operador determinado.

Si aplicamos el operador NOT al contenido de un registro, lo que obtenemos es el "complemento" de ese registro, es decir, cambiamos sus "unos" por "ceros" y sus "ceros" por "unos". A esta operación se la denomina "complementar un registro", y utilizamos para ello la instrucción CPL del microprocesador.

Veamos un ejemplo: Supongamos que el registro acumulador contiene el número:

```
Ø 1 1 Ø 1 Ø 1 Ø
```

Que se podría escribir como "6Ah" en hexadecimal (ver capítulo referente a los sistemas de numeración). Si lo complementamos, obtenemos:

```
1 Ø Ø 1 Ø 1 Ø 1
```

Que podría escribirse como "95h" en hexadecimal. Hemos cambiado los "ceros" por "unos" y los "unos" por "ceros". El número "95h" es el complementario de "6Ah" porque si los sumamos, obtenemos "FFh", que a su vez, es el mayor número posible (todos son "unos").

De esta forma, sería posible construir una "tabla" para la operación NOT en hexadecimal. Esta tabla la hemos representado en la FIGURA 3. Se puede observar que si sumamos cualquier número, con el que resulta de aplicarle el operador NOT, obtenemos "Fh", por eso son "complementarios".

OPERADOR OR

Se trata de una de las operaciones que se usan para definir nuestra álgebra de Boole, y es equivalente a la "suma" en el sentido de que satisface las mismas propiedades algebraicas.

Cuando operamos dos elementos de nuestro conjunto mediante este operador, obtenemos un "1" si al menos, uno de ellos es "1", o si lo son ambos; y obtenemos "Ø" en cualquier otro caso. La tabla de verdad del operador "OR" es la siguiente:

```
Ø OR Ø = Ø
Ø OR 1 = 1
1 OR Ø = 1
1 OR 1 = 1
```

Veamos un ejemplo: Supongamos que el acumulador contiene el número:

```
Ø 1 Ø Ø Ø 1 1 Ø
```

Es decir, "46h". Y le hacemos un "OR" con el número:

```
1 1 1 Ø Ø Ø 1 1
```

Es decir, "E3h". El resultado sería el número:

```
1 1 1 Ø Ø 1 1 1
```

Que se representa en hexadecimal como "E7h". Vemos que hemos puesto un "1" en los lugares donde había "1" en cualquiera de los dos números y "Ø" en los lugares donde ambos números tenían un "Ø". La tabla para este operador en hexadecimal se puede ver en la FIGURA 4.



## PROGRAMA 1

```
  10 REM   ** SCROLL LATERAL **
  20 REM   Curso C/M MICROHOBBY
  30 REM
  40 REM
  50 CLEAR 54999: LET d=55000
  60 DEF FN a(a$,n)=16*(CODE a$(
n)-48-7*(a$(n)>"9"))+(CODE a$(n+
1)-48-7*(a$(n+1)>"9"))
  70 FOR f=1 TO 6
  80 READ a$,s: LET c=0
  90 FOR n=1 TO LEN a$-1 STEP 2
 100 LET a=FN a(a$,n): POKE d,a
 110 LET d=d+1: LET c=c+a
 120 NEXT n
 130 IF c<>s THEN PRINT "<ERROR>
en la linea: ";300+10*f: STOP
 140 NEXT f
 200 REM DEMOSTRACION
 210 PRINT '''" DEMOSTRACION DE
SCROLL LATERAL "'''" PULSE:"'''"
Q Para SCROLL a Izquierda"'''" P
 Para SCROLL a Derecha"
 220 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175
 230 POKE 23658,0
 240 IF INKEY$="q" THEN RANDOMIZ
E USR 55000
 250 IF INKEY$="p" THEN RANDOMIZ
E USR 55030
 260 GO TO 230
 300 REM CODIGO MAQUINA
 310 DATA "21FF570EC00620A7CB16"
,1011
 320 DATA "2B10FB300B22B05CDD2A"
,934
 330 DATA "B05CDDCB20C60D20E8C9"
,1400
 340 DATA "2100400EC00620A7CB1E"
,741
 350 DATA "2310FB300B22B05CDD2A"
,926
 360 DATA "B05CDDCBE0FE0D20E8C9"
,1640
```

Para que la pantalla se desplace un pixel a la izquierda, y:

```
RANDOMIZE USR 55030
```

Para que lo haga hacia la derecha. Para salvar el código en cinta, puede utilizar:

```
SAVE "Sroll" CODE 55000,60
```

Y para cargarlo:

```
CLEAR 54999:
LOAD "Scroll" CODE 55000
```

El PROGRAMA 1 incluye una demostración sobre la forma de utilizar estas rutinas. Quien esté interesado en usarlas en sus programas, puede mirar atentamente las líneas 240 y 250 que resultan suficientemente ilustrativas.

Para adaptar las rutinas a la versión de 16 K, se deben realizar las siguientes modificaciones en el PROGRAMA 1:

```
 50 CLEAR 31999: LET d=32000
 240 IF INKEY$="q" THEN RANDOMIZ
E USR 32000
 250 IF INKEY$="p" THEN RANDOMIZ
E USR 32030
```

En este caso, habrá que salvar las rutinas con:

```
SAVE "Scroll"CODE 320000,60
```

Y volverlas a cargar con:

```
CLEAR 31999:
LOAD "Scroll" CODE 32000
```

Antes de seguir leyendo, le pedimos que cargue y ejecute el PROGRAMA 1. Ponga mucho cuidado para no equivocarse a partir de la línea 300, ya que los errores en código máquina suelen tener consecuencias desastrosas.

¿Ya ha ejecutado el programa? Asombroso ¿no? Tenemos pensado otro que hace el Scroll arriba y abajo, ya se lo contaremos.

Ahora vamos a intentar introducirnos en el estudio del código máquina partiendo desde la base más elemental, para que incluso quien no tenga ni la más remota idea de lo que es esto pueda seguirnos. Si no es su caso y es usted capaz de entender sin problemas las explicaciones de los cuatro primeros capítulos no es necesario que lea lo que sigue, aunque tal vez pueda aclararle algunos conceptos.

### Manejando una calculadora

Suponemos que todos nuestros lectores han manejado alguna vez una calculadora de bolsillo. El conjunto formado por una calculadora de bolsillo, la persona que la maneja, un lápiz y un papel, pueden ser un símil bastante aproximado de lo que es un ordenador.

Imaginémonos a un amigo con una calculadora y un lápiz; esto equivale más o menos al microprocesador o CPU. Por otro lado está el papel, que equivale a la memoria. Nuestro amigo puede usar el papel para apuntar resultados o datos intermedios de los cálculos, pero nosotros podemos usarlo también, para apuntarle a él los cálculos que queremos que realice. De esta forma, el papel (la memoria) cumple una doble función, por un lado sirve para que el microprocesador (nuestro amigo) anote datos, y por otro lado, sirve para que nosotros le anotemos las instrucciones que tiene que seguir (el programa).

Nuestro amigo no tiene ni idea de como se maneja una calculadora, así que tendremos que decirle, una por otra, las teclas que tiene que pulsar. Podemos decirle: "pulsa la segunda tecla de la tercera fila" o simplemente: "pulsa (2,3)"; esto sería "código má-

quina". Pero también podemos decirle: "pulsa la tecla 5, luego la tecla «por» y luego la tecla 7"; esto sería "Assembler".

Vamos a "programar" en "Assembler" a nuestro amigo, para que nos calcule el cuadrado de 5 por 7 y no escriba el resultado en un recuadro de la hoja de papel al que denominamos "archivo de presentación visual". La calculadora puede ser la representada en la FIGURA 2.

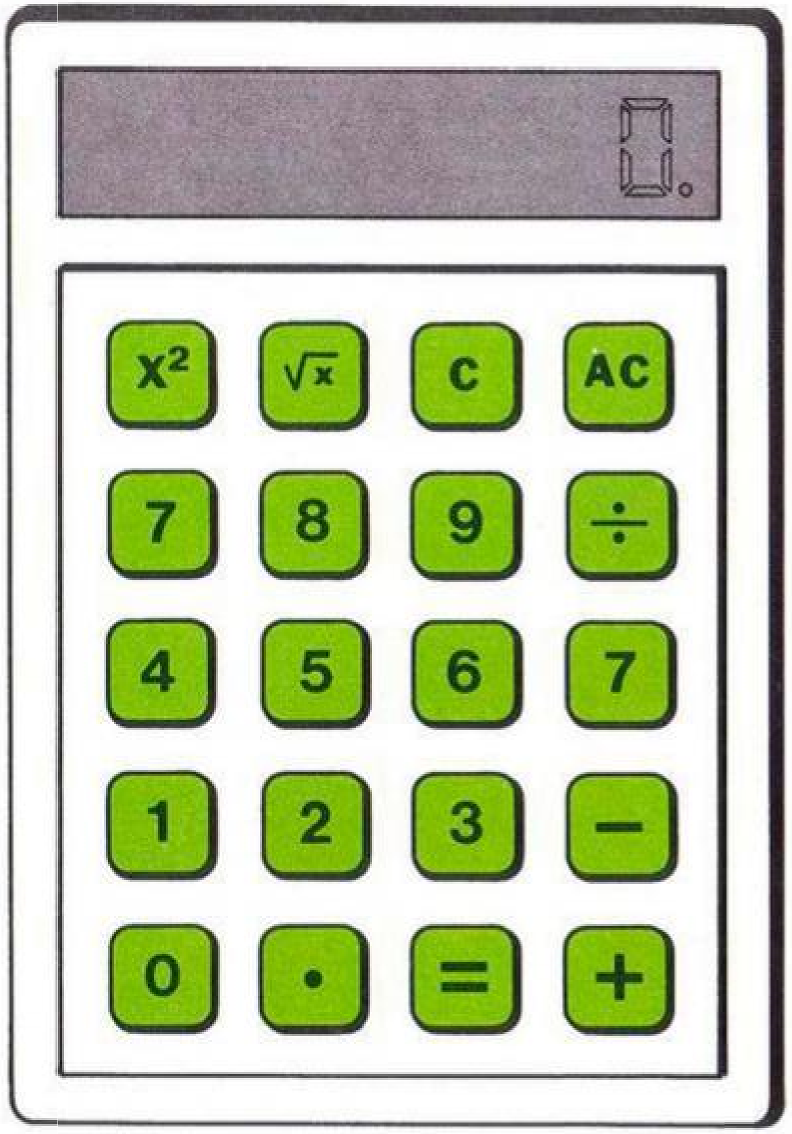
El programa podría quedar más o menos así:

```
10 Pulsa "AC"
20 Pulsa "5"
30 Pulsa "POR"
40 Pulsa "7"
50 Pulsa "="
60 Pulsa "CUADRADO"
70 Escribe Resultado
80 Fin
```

Este programa se lo anotamos en el papel, y le damos la orden de que lo ejecute. Al final, él nos escribe el resultado en el papel.

Podemos sacar aún más partido a nuestro ejemplo. Supongamos que nuestro amigo supiera manejar perfectamente la calculadora, en ese caso, nos bastaría con decirle: "Calcula el cuadrado de 5 por 7 y anota el resultado". En este caso, estaríamos usando un "lenguaje de alto nivel". El Basic es un lenguaje de alto nivel, y lo podemos usar gracias a que nuestro ordenador tiene un "intérprete", lo que hace que "sepa" manejar perfectamente el microprocesador.

Vamos a estudiar detenidamente el proceso. Primero pulsamos "AC", con lo que se borran los anteriores contenidos de la calculadora. A




**La calculadora de nuestro amigo el microprocesador.**

continuación, pulsamos la tecla "5", con lo que aparece en pantalla el número cinco. La pantalla de la calculadora es un registro, y lo que hemos hecho ha sido cargar este registro con el número cinco. Luego definimos la operación a realizar, y cargamos el registro con el segundo operando (siete). Al pulsar "=" se realiza la operación y el resultado aparece de nuevo en ese registro. Finalmente, al pulsar "CUADRADO", elevamos al cuadrado el contenido del registro, y el resultado nos vuelve a aparecer en el mismo.

La pantalla de una calculadora va acumulando los resultados de todas las operaciones que vamos realizando, por eso, podemos llamarla "registro ACUMULADOR". Todos los microprocesadores tienen un registro acumulador, pero a diferencia de las calculadoras, tienen también otros registros que nos pueden servir para diversos fines.

A pesar de tener muchos registros, no son suficientes para almacenar el enorme volumen de datos que maneja un ordenador, por ello se recurre a un sistema de almacenamiento externo que se denomina "memoria" y cumple la misma función que el papel de nuestro ejemplo, sirve para almacenar tanto programas, como datos.

El "papel" que utilizamos como memoria está "cuadriculado" y nuestro microprocesador no sólo tiene "lápiz", sino también "goma" por lo que puede escribir y borrar en cualquiera de las cuadrículas; pero sólo borra una cuadrícula cuando tiene que escribir otro dato en ella. Las cuadrículas, a su vez, están numeradas, por lo que en cada caso, se puede acceder a una de ellas en concreto.

Existen más diferencias entre la calculadora y el microprocesador. Este último no realiza las mismas operaciones que una calculadora. Es cierto que puede sumar y restar, pero puede realizar también otro tipo de operaciones como incrementar un registro (sumarle 1), decrementarlo (restarle 1) rotarlo, y fundamentalmente, realizar lo que se denomina "operaciones lógicas" (AND, OR, NOT, EXOR). Además, nos informa continuamente de ciertas características del dato que contiene el acumulador, por ejemplo, nos dice si es cero, si es negativo, y otros cuya utilidad se irá viendo más adelante.

No obstante, la diferencia fundamental entre nuestro ejemplo y un verdadero microprocesador es que este último no trabaja en base 10 (decimal), sino en binario. Afortunadamente, no necesitamos trabajar siempre con números binarios (que son sumamente incómodos) y podremos utilizar números en base 16 (hexadecimales).

Es muy importante adquirir cierta soltura en el manejo de la numeración hexadecimal, por ello, hemos dedicado un capítulo entero a este tema. Con bastante frecuencia, tendremos que convertir números decimales a hexadecimales o viceversa. Para esto se pueden usar los métodos descritos en el citado capítulo, pero resulta bastante tedioso, así que hemos desarrollado un programa que hace ese trabajo por nosotros. Este programa se encuentra en la página 11 del curso (MICROHOBBY número 43). Existen también, calculadoras de bolsillo capaces de operar en estas bases, y representan una gran ayuda a la hora de programar en código máquina.

También hemos dedicado un capítulo a describir el microprocesador Z-80 con el mayor detalle posible, ya que su conocimiento es imprescindible para programarlo. Sería algo así como el "manual" de la calculadora.

Finalmente, y antes de empezar a estudiar las instrucciones, hemos dedicado un capítulo a describir la forma en la que se debe elaborar un programa, independientemente del lenguaje utilizado.

## Algebra de Boole

Prácticamente todos los instrumentos matemáticos que se utilizan en la programación de un pequeño ordenador como el Spectrum, forman parte del bagaje cultural de cualquier persona medianamente formada. Excepto, quizá, el álgebra de Boole.

Tal vez por ser de aparición relativamente reciente, tal vez por su escasa utilidad práctica en la realidad habitual, el caso es que el álgebra de Boole no ha sido incluida en el programa de estudios hasta fecha reciente; y lo ha sido dentro de la asignatura de "matemáticas comunes" del C.O.U. desgraciadamente, una de las "Marías".

Programando en Basic, hemos hecho uso de algunos conceptos provenientes del álgebra de Boole; cuando utilizábamos en las sentencias IF... THEN, los operadores OR, AND y NOT para expresar conjunciones o disjunciones lógicas. Al programar en Assembler o código máquina, haremos un uso mucho más profundo y preciso de estos operadores, así como del operador EXOR que no se utiliza en Basic.

Es tan frecuente (y útil) utilizar operadores lógicos en Assembler, como lo puede ser utilizar la suma y la resta en una calculadora de bolsillo. Por ello, es imprescindible tener un cierto conocimiento del álgebra de Boole; que por otro lado, es sumamente sencilla de aprender.

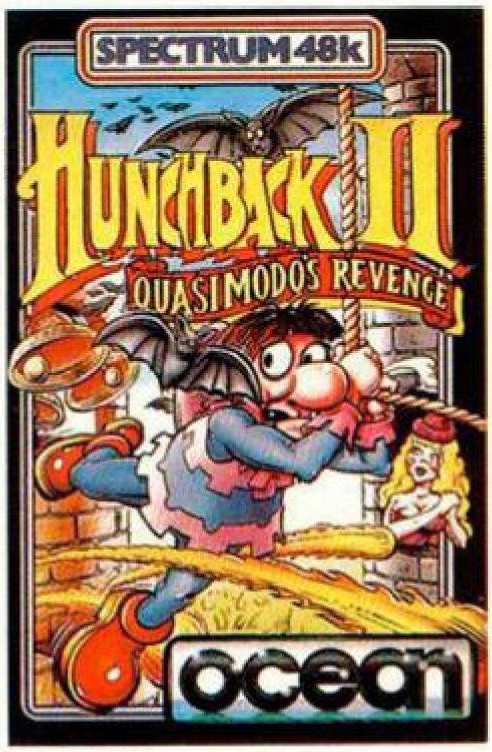
Dado que éste no pretende ser un manual de matemática moderna, no entraremos a definir formalmente lo que constituye un álgebra de Boole. Bástenos saber que un álgebra de Boole se puede construir allá donde tengamos un conjunto de elementos que puedan tomar dos valores (en nuestro caso, "0" y "1") y definamos una relación de equivalencia ("ser igual a") y dos operaciones internas al conjunto, que cumplan una serie de propiedades, similares a las que cumplen la suma

# Una oferta que vale por... 5

## AL REALIZAR TU SUSCRIPCION A MICROHOBBY RECIBIRAS TOTALMENTE GRATIS UNA SELECCION DE FABULOSOS PROGRAMAS ELEGIDOS ENTRE LOS DE MAYOR EXITO DE TU FIRMA DE SOFTWARE FAVORITA: US GOLD

**Blue Max**
Tienes los controles de un avión de la primera guerra en tus manos. Derriba al enemigo, bombardea blancos, destruye tanques y emplazamientos de armas.



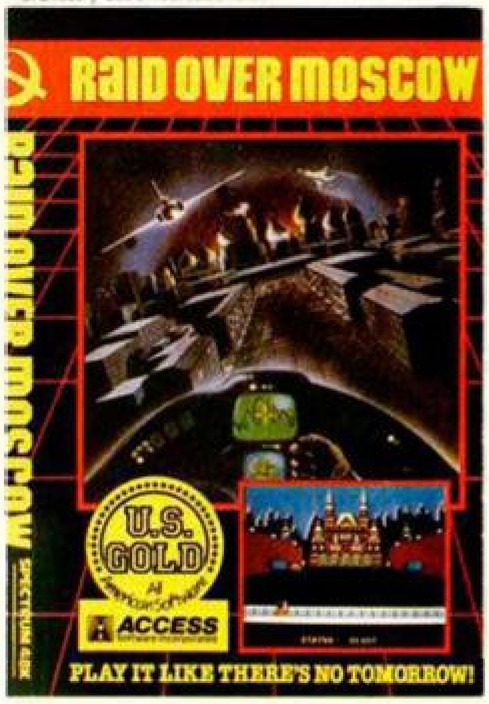




**Hunchback II**
Quasimodo busca a Esmeralda prisionera en el castillo. Durante las seis primeras imágenes debes accionar la campana gigante, hasta llegar a la liberación de Esmeralda.

**Flak**
En el año 2096 el universo está controlado por ordenadores. Una fuerza siniestra que quiere eliminar la libertad. Tu misión como piloto guerrero es atacar y destruir esta amenaza.




**Raid Over Moscow**
Defiende a USA y Canadá del ataque nuclear que ha lanzado Rusia contra ellos. Con tu escuadrilla habrás de hacer un viaje lleno de peligros hasta llegar al mismísimo Kremlim y destruir las bases de lanzamiento soviéticas. Gráficos y acción sensacionales.

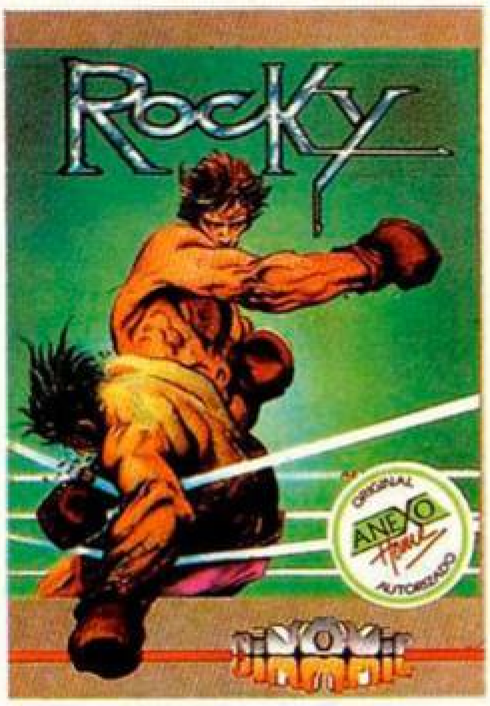



Estos cinco programas, que han encabezado listas de éxitos de toda Europa durante este año, están contenidos en **dos cintas** de cassette para Spectrum. De esta forma van a aparecer en breves días en el mercado inglés, como gran oferta de Navidad de la firma US GOLD, bajo el nombre genérico de **«Arcade Hall of Fame»**. En España se pondrán a la venta al precio de 2.500 ptas., aunque el valor total de los cinco programas es muy superior.

**Microhobby** Semanal te regala ahora estos cinco estupendos programas, al realizar tu suscripción, sólo hasta el próximo 31 de diciembre.

Lo increíble de esta oferta-regalo es que su valor es casi el 50 por 100 del precio real de suscripción.

Suscribirse a **Microhobby** es el sistema ideal para recibir nuestra Revista puntualmente y para ahorrar mucho dinero en su precio. Si consideras, además, el valor de los programas que recibes de regalo, no nos cabe duda de que preferirás recibir en tu casa **Microhobby** el próximo año.




**Rocky**
Vive la emoción de un encarnizado combate de boxeo, con Rocky. Tendrás que pelear duro para conseguir el campeonato del mundo; sólo los puños de tu oponente se interponen en tu camino hacia el triunfo final.

***ENVIA HOY MISMO EL CUPON QUE ESTA COSIDO AL FINAL DE LA REVISTA Y ELIGE LA FORMA DE PAGO QUE TE RESULTE MAS FAVORABLE***

# Es usted capaz de tomar el relevo del General Montgomery...?

## Juegos & ESTRATEGIA

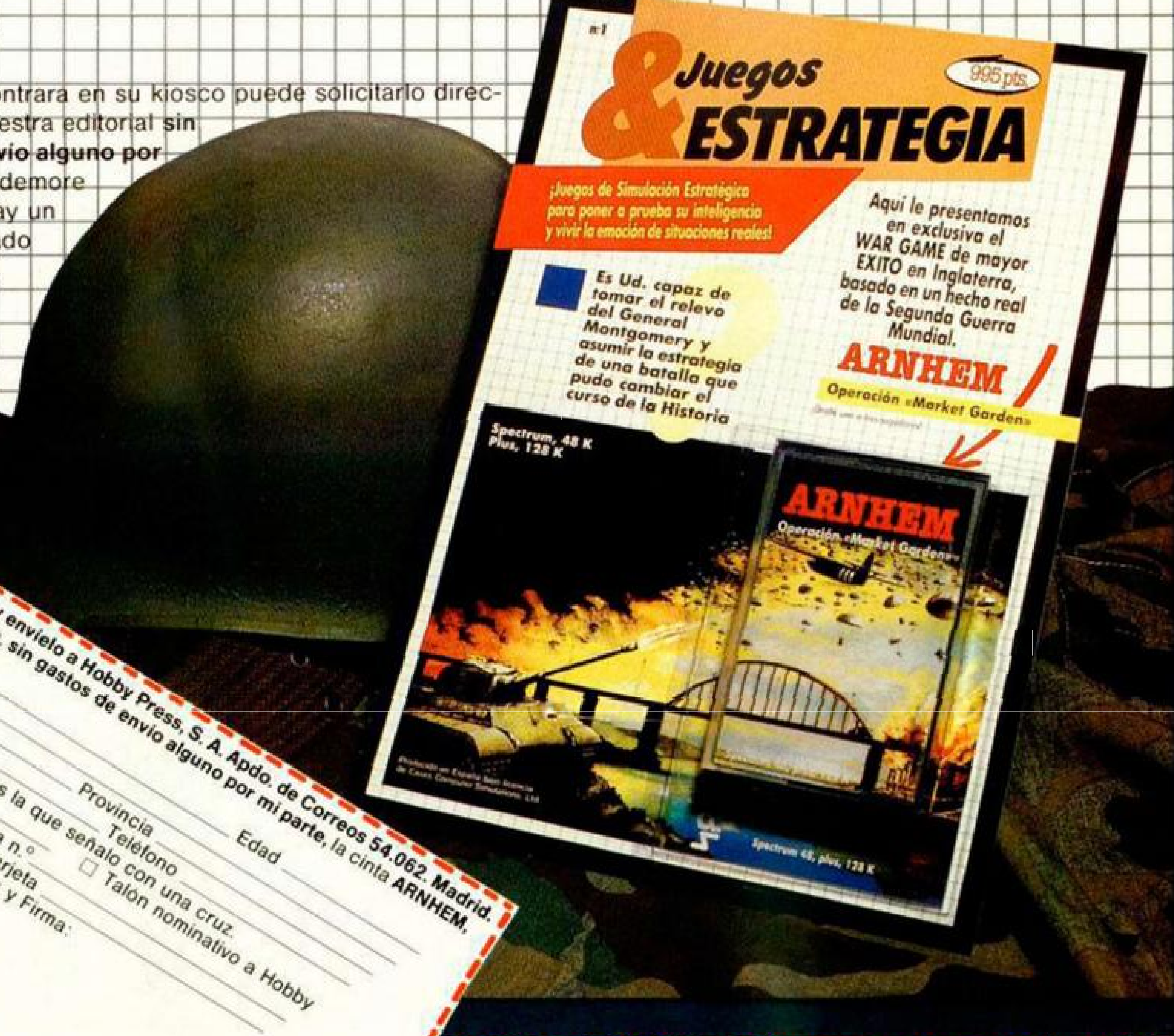
Ya está a la venta!

**le presenta en exclusiva el WAR GAME, para Spectrum, de mayor éxito en Inglaterra:**

## ARNHEM

**(operación «*Market Garden*», basada en un hecho real de la Segunda Guerra Mundial)**

Si no lo encontrara en su kiosco puede solicitarlo directamente a nuestra editorial **sin gastos de envío alguno por su parte.** No demore su pedido, hay un número limitado de cassettes.




**Recorte o copie este cupón y envíelo a Hobby Press, S. A. Apdo. de Correos 54.062. Madrid.**
Deseo recibir en mi domicilio, **sin gastos de envío alguno por mi parte,** la cinta **ARNHEM,** al precio de 995 pesetas.
Nombre ______ Edad ______
Dirección ______
Localidad ______ Provincia ______
Código ______ Teléfono ______
La forma de pago elegida es la que señalo con una cruz.
☐ Tarjeta Visa n.º ______ ☐ Talón nominativo a Hobby Press, S.A.
☐ Giro Postal n.º ______
Fecha de caducidad de la tarjeta ______ Fecha y Firma:

**HOBBY PRESS, S.A. Editamos para gente inquieta.**

# COMPILADOR DE TEXTOS

A. G. GARRE

**Cuando hemos hecho un programa a nuestro gusto inmediatamente nos damos cuenta que nos gustaría hacer partícipes a los amigos de él, para que lo usen y les resulte de utilidad. Pero enseguida caemos en la cuenta: ¿cómo funciona?, lo que está clarísimo para nosotros no lo está para los demás, por lo que decidimos que se deben adjuntar unas instrucciones. Y qué mejor que unas instrucciones en pantalla al iniciar el programa y alguna pantallita de ayuda en los pasajes más complicados.**



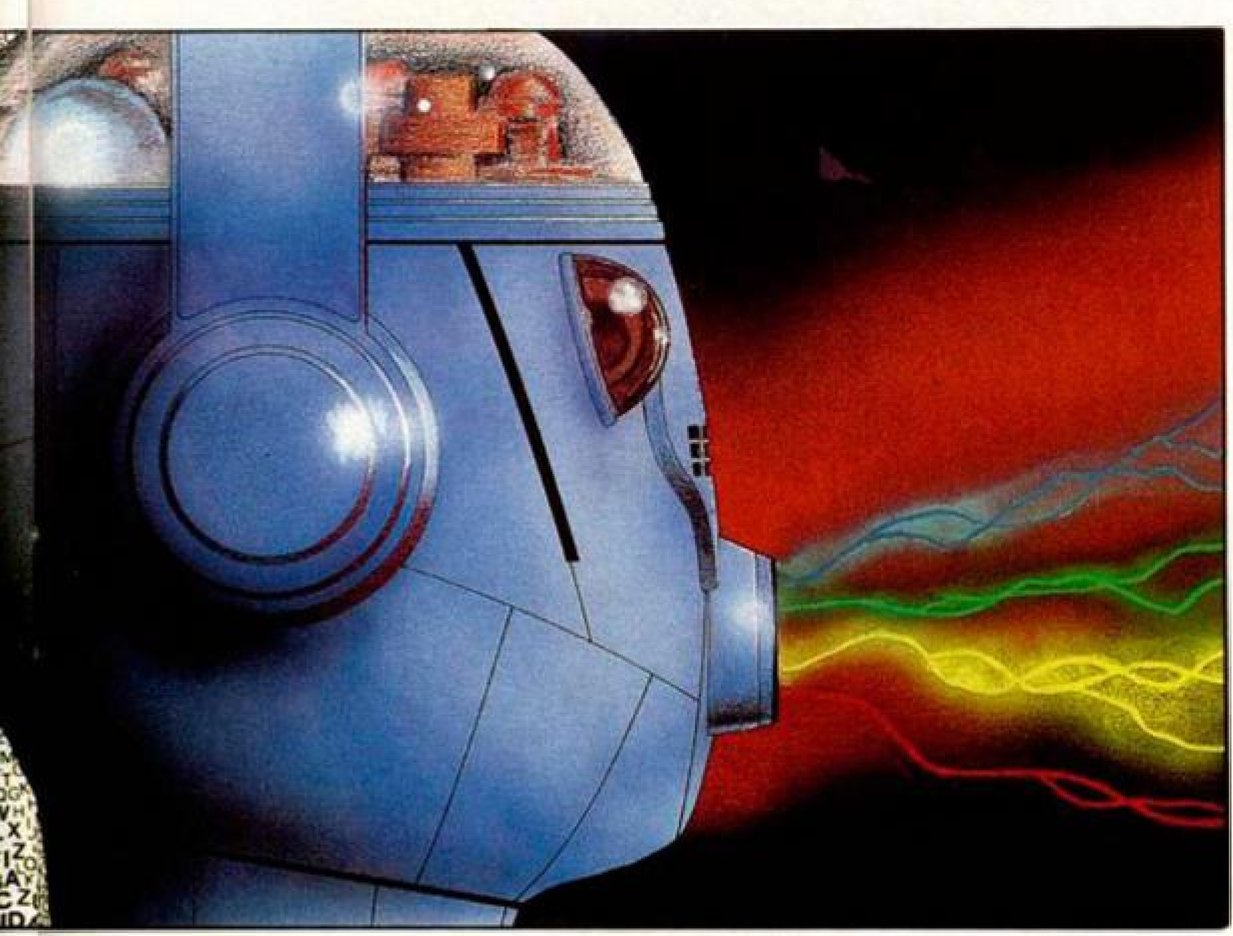
Ponemos manos a la obra y empezamos con el consiguiente
13 PRINT "El presente programa realiza las ecuaciones..."

En primer lugar, si no lo teníamos previsto, no tenemos las líneas de programa suficientes al principio para escribir todas las explicaciones que queremos dar. Si por el contrario sí lo teníamos previsto o empleamos GOSUB hacia el final del programa, tenemos sitio suficiente, pero las 32 columnas de nuestro televisor no son múltiplo exacto de nuestras palabras y al ver que las instrucciones aparecen decapitadas, tenemos que editar una y otra vez para que la cosa quede legible. Esta tarea es bastante penosa, pues cuando se corrige una letra en la siguiente fila lo que antes estaba bien, ahora aparece cortado. Vuelta a empezar EDIT y clic, clic, clic hasta que llegamos a la palabreja. Si por curiosidad controlamos el tiempo, veremos que se nos ha pasado una hora para dos o tres pantallas completas. Hasta que por fin, más cansados que satisfechos damos por concluida la labor abreviando lo más posible. Se pueden emplear trucos, como empezar a escribir a principio de línea, y así se preveé donde se cortarán las palabras. de todos modos los listados pierden claridad con las instrucciones, se alargan y después cuando queremos corregir algo en el programa, no vemos el grano entre tanta paja.

Para paliar en lo posible este problema recomendamos a todas aquellas personas que saquen mucha información por pantalla y quieran dejar claros y bien estructurados sus programas, que utilicen éste titulado COMPILADOR DE PANTALLA, por que todo aquello que se escriba en la pantalla se «compila» en forma de una rutina en código máquina que no precisa el BASIC para ejecutarse, no se «ve» en el listado, y que basta RANDOMIZE USR 42000–63000 para obtener una pantalla completa de instrucciones de forma instantánea y que además gasta menos memoria que sus homólogas en basic.

**TODAS LA LETRAS MAYUSCULAS SUBRAYADAS DEBERAN TECLEARSE EN MODO GRAFICO.**

```
   1 REM * COMPILADOR PANTALLA *
                © A.G. GARRE 85
   5 CLEAR 39000
   7 FOR n=39500 TO 39593: READ
a: POKE n,a: NEXT n
   8 DATA 1,114,159,33,63,156,17
,64,156,35,126,254,255,40,65,254
,32,40,246,62,22,2,3,229,237,82,
229,203,61,203,61,203,61,203,61,
203,61,203,68,40,2,203,221,203,7
6,40,2,203,229,125,2,3,225,203,1
69,203,181,203,173,125,2,3,225,1
26,2,3,35,126,254,255,40,8,254,3
2,40,169,2,3,24,242,62,255,2,3,2
37,67,66,159,72,237,67,67,159,20
1
  10 DIM A(20): DIM B(20): LET C
ONT=1
  20 LET CA=40000: FOR N=ca+720
TO ca+735: READ a: POKE N,a: NEX
T N
  21 DATA 1,195,130,62,32,33,64,
156,119,35,11,203,120,32,249,201
  30 FOR N=ca+800 TO ca+817: REA
D A: POKE N,A: NEXT N
  40 DATA 62,2,205,1,22,33,114,1
59,126,254,255,40,4,215,35,24,24
7,201
  50 FOR n=40750 TO 40761: READ
a: POKE n,a: NEXT n
  51 DATA 1,1,0,17,16,164,33,96,
159,237,176,201
  70 BORDER 1: PAPER 0: INK 7: C
LS
 110 FOR N=0 TO 7: READ a: POKE
USR "a"+n,a: NEXT n
 120 DATA 0,0,0,0,0,24,60,126
 200 RANDOMIZE USR 40720: POKE 4
0704,255: POKE 23658,0
 220 POKE 23561,10: POKE 23562,1
: LET u=0
 230 INPUT "": PRINT #0;"STOP pa
ra menu"
 250 LET CCP=CA+818
 890 REM [illegible]
 895 PRINT AT 0,0; OVER 1;"B"
 900 FOR v=0 TO 21: FOR h=0 TO 3
1
 910 POKE 23560,0: PRINT AT v,h;
 OVER 1;"B"
 920 LET U=PEEK 23560
 930 IF u=0 THEN PRINT AT v,h; O
VER 1;"B";AT v,h; OVER 1;"B": GO
 TO 920
 940 IF u=226 THEN PRINT AT v,h;
 OVER 1;"B": PRINT AT V,H; OVER
1;"B": POKE 23561,35: POKE 23562
,5: GO SUB 6000
 950 POKE 23560,0: LET e$=CHR$ u
 960 IF U<32 THEN GO SUB 2000
1000 IF CODE E$>=32 THEN PRINT A
T v,h;e$: POKE (CA+(U*32)+H),COD
E E$
1001 BEEP .0005,45
1002 IF h=31 AND V=21 THEN LET H
=H-1
1005 IF h=31 THEN GO TO 1100
1010 PRINT AT v,h+1; BRIGHT 1; O
VER 1;"A";
1100 NEXT h: NEXT v
2000 REM [illegible]
2001 IF u=12 THEN POKE 23560,0:
GO SUB 9500: POKE (CA+(V*32)+H),
32: POKE ((CA+(V*32)+H)-1),32: P
RINT AT v,h-1;"  ": LET h=h-2: L
ET e$="": GO TO 1000
2010 IF u=13 THEN LET e$="": PRI
NT AT v,h; OVER 1;" ": LET v=v+1
: LET h=-1: IF v>21 THEN LET v=2
1: GO TO 1100
2020 IF u=10 THEN : LET e$="": P
RINT AT v,h; OVER 1;" ": LET v=v
+1: LET h=h-1: IF v>21 THEN LET
v=21: GO TO 1001
2030 REM IF u=11 THEN LET e$="":
 PRINT AT v,h; OVER 1;" ": IF v<
0 THEN LET v=0: GO TO 1001
2040 IF u=11 THEN : LET e$="": P
RINT AT v,h; OVER 1;" ": LET v=v
-1: LET h=h-1: IF v<0 THEN LET v
=0: GO TO 1001
2050 IF u=9 THEN LET e$="": PRIN
T AT v,h; OVER 1;" ": IF h>30 TH
EN LET h=30: GO TO 1001
2060 IF u=8 THEN LET e$="": PRIN
T OVER 1;AT v,h-1;"  ": LET h=h
-2: IF h<0 THEN LET h=-1: GO TO
1001
2065 IF u=6 AND PEEK 23658=0 THE
N POKE 23658,8: RETURN
2066 IF u=6 AND PEEK 23658=8 THE
N POKE 23658,0: RETURN
2070 RETURN
5000 RANDOMIZE USR 39500: REM [illegible]
5001 LET CCP=(PEEK 40771*256+PEE
K 40770)
5110 CLS : GO SUB 5800
5200 REM [illegible]
5210 INPUT "COMIENZO ?",CR
5220 IF CR<42000 OR CR>63000 THE
N GO TO 5210
5230 RANDOMIZE CR: POKE 40754,PE
EK 23670: POKE 40755,PEEK 23671:
 RANDOMIZE (CR+18): POKE 40806,P
EEK 23670: POKE 40807,PEEK 23671
: REM [illegible]
5240 LET A(CONT)=CR: LET B(CONT)
=CCP-(CA+800)
5250 RANDOMIZE (B(CONT)+1): POKE
 40751,PEEK 23670: POKE 40752,PE
EK 23671
5251 LET CCP=CCP+1
5260 RANDOMIZE USR 40750
5600 LET CONT=CONT+1
5610 CLS : GO SUB 5800: INPUT ""
: PRINT #0;"Ver Editor Salvar":
 POKE 23658,8
5620 IF INKEY$="E" THEN CLS : GO
 TO 200
5625 IF INKEY$="S" THEN GO TO 57
00
5630 IF INKEY$<>"V" THEN GO TO 5
620
5635 PAUSE 30
5640 INPUT "QUE NUMERO ? [illegible]
";NUM: IF NUM<1 OR NUM>20 THEN G
O TO 5640
5650 IF A(NUM)=0 THEN INPUT "":
PRINT #0; FLASH 1;"NO HAY RUTINA
 PRESENTE": PAUSE 75: GO TO 5610
5660 CLS : RANDOMIZE USR A(NUM):
 PAUSE 0: GO TO 5610
5700 REM [illegible]
5710 INPUT "NOMBRE ", LINE A$: I
F LEN A$>10 OR LEN A$<1 THEN GO
TO 5710
5720 INPUT "COMIENZO ",COMI: INP
```



área de 40000 a 40703 que es donde se «escribe» el texto además de escribirse en pantalla.

El borrado se efectúa sobreescribiendo el carácter 32, que es el espacio en blanco, para iniciar una nueva pantalla.

30-40 Carga la rutina lectora de los datos. Son 18 bytes que van con cada pantalla y son los que leen e interpretan desde el byte 19 hasta que encuentran el carácter ASCII 255 que se emplea como marcador del fin de los datos. La llamada USR se ha de efectuar necesariamente a la posición del primer byte de esta sencilla subrutina. Es pues imprescindible tomar nota de la posición de este primer byte.

50-51 Carga la subrutina que moverá el bloque de lector de datos + datos + 255, desde la posición en la que la «escribe» el programa (40800 hasta...), hasta la posición que desea el usuario.

### Descripción del programa

Fundamentalmente consiste en un editor de texto con el que es posible escribir a voluntad una pantalla con instrucciones, menús, etc., colocando todo el contenido de dicha rutina a la que se llama con USR desde Basic. Dicha rutina AHORRA en la mayoría de los casos (depende de la longitud de texto) memoria, resulta MAS FACIL confeccionar pantallas de instrucciones que con sentencias PRINT, PRINT AT, y se clarifican notablemente los listados puesto que al estar totalmente en C/M no se «ven» las instrucciones.

```
UT "ULTIMO",ULTI: LET BYT=(ULTI-
COMI)+1
5730 CLS : PRINT AT 9,5;"NOMBRE
    ";A$;AT 10,5;"COMIENZO
    ";COMI;AT 11,5;"ULTIMO BYTE
    ";ULTI;AT 12,5;"BYTES A SALV
AR   ";BYT
5740 PRINT #0;"ES CORRECTO?"
5750 IF INKEY$="N" THEN GO TO 57
00
5755 IF INKEY$="S" THEN GO TO 57
60
5759 GO TO 5750
5760 PAUSE 0: INPUT "": SAVE A$C
ODE COMI,BYT
5770 PRINT AT 19,0; FLASH 1;"
REBOBINA PARA VERIFICACION    ":
VERIFY ""CODE COMI,BYT: CLS : GO
 TO 200
5800 REM [illegible]
5810 PRINT INVERSE 1;"NUM. COMIE
NZO  LONG.     ULTIMO    ": FOR N=1
 TO 20
5820 PRINT N;".....";TAB 7;A(N);T
AB 16;B(N);TAB 24;A(N)+B(N): NEX
T N
5830 PRINT INVERSE 1;"       ENTR
E 42000 Y 63000        "
5840 RETURN
5999 REM [illegible]
6000 INPUT "": POKE 23658,8: PRI
NT #0;"Compilar Editor Iniciar [illegible]
er": PAUSE 0
6010 IF INKEY$="C" THEN GO TO 50
00
6020 IF INKEY$="E" THEN LET e$="
": LET u=32: POKE 23560,32: INP
UT "": PRINT #0;"[illegible] para menu"
: RETURN
6030 IF INKEY$="U" THEN GO TO 56
10
6040 IF INKEY$="I" THEN CLS : GO
 TO 200
6050 GO TO 6000
8999 REM [illegible]
9500 IF h=0 AND v<>0 THEN PRINT
AT v,h;"A": PRINT AT v,h; OVER 1
;"A": LET h=31: LET v=v-1
9501 IF h=0 AND v=0 THEN BEEP .5
,-10: GO TO 895
9510 RETURN
```

En el programa se ha previsto que se puedan almacenar hasta 20 pantallas aunque es posible, si fuera necesario, almacenar más, dependiendo de la memoria que ocupe cada una.

La razón de ser 20 pantallas es por considerar que en la mayoría de los casos serían suficientes y por visualizarlas de una forma cómoda en una sola «pantalla». Si se precisaran más pantallas, y hubiera sitio suficiente entre 42000 y 63000, toma nota cuidadosa del comienzo de las primeras 20, BREAK al programa, da RUN y haz corresponder la pantalla 21 con la una de esta segunda tanda, respetando, como es lógico, el sitio donde se han ubicado anteriormente las 20 primeras para que no se solapen las unas con las otras. Pueden hacer tantas tandas como quepan en memoria. Para salvarlas debe hacerse de una sola vez, desde 42000 (o la primera dirección que se haya puesto) hasta el último Byte de la última pantalla.

### Estructura

Consta el programa de 4 partes fundamentales:

*1-120 INICIALIZACION*

7-8 Carga la rutina de compilación del texto de la que se adjunta listado en ensamblador. Se ha realizado en código máquina pues en basic tardaba unos 28 segundos en realizarse mientras que así tarda aproximadamente 0.04 segundos.

20-21 Carga la rutina de borrado del área de 40000 a 40703 que es donde se «escribe» el texto además de escribirse en pantalla.

*890-2070 CONTROL DEL EDITOR DE TEXTO*

Permite el borrado y mover el cursor por la pantalla para corregir, (no justifica ni inserta) del mismo modo que lo hace el Spectrum, o como lo haría una máquina de escribir, es decir es un editor de texto, no un procesador.

Está escrito en BASIC, y es bastante rápido aunque si se desea mayor rapidez puede alterar la línea 220 con otros valores en las posiciones de las variables del sistema REPDEL y REPPER 23561 y 23562 o valores de repetición y retardo.

*5000-6000 COMPILA Y GUARDA LOS DATOS*

La línea 5000 llama a la rutina de compilación. El programa BASIC que hace exactamente lo mismo sería:

```
5000 FOR N=40000 TO 40703
5020 IF PEEK<=32 THEN GOTO
     5100
5030 POKE CCP,22
5040 POKE   CCP+1,INT((N-CA)/
     32)
5050 POKE   CCP+2,(N-CA)-(INT
     ((N-CA)/32)*32)
5060 POKE  CCP+3,PEEK  N:LET
     CCP=CCP+4
5070 IF  PEEK  (N+1)<=32  THEN
     GOTO 5100
5080 LET   N=N+1:POKE   CCP,
     PEEK N
5090 LET      CCP=CCP+1:GOTO
     5070
5100 NEXT N:POKE CCP, 225
```

Pueden sustituirse las líneas 5000 y

5ØØ1 por el programa anterior y comprobar que ambos hacen exactamente lo mismo. La única diferencia está en el tiempo.

El resto, hasta la línea 6ØØØ, gestiona el archivo de una forma sencilla con las matrices A(2Ø) y B(2Ø) que se dimensionan en la línea 1Ø. Al guardarse el comienzo y la longitud de las pantallas en matrices hay que tener cuidado con RUN, ya que esta instrucción inicializa dichas matrices y aunque a lo que ya está compilado no le afecta para nada, es decir, que las pantallas no se pierden, lo que sí perdemos es el comienzo de cada una de las pantallas y si no lo hemos anotado no sabremos dónde empiezan y por consiguiente no podremos usarla.

*6ØØØ-6Ø5Ø ZONA DEL MENU DE OPCIONES*

La letra clave de acceso es la primera en inverso. Es decir, si queremos compilar debemos pulsar la C, si es Ver las pantallas la V.

Cuando dice STOP para menú, se refiere a la palabra clave STOP que se obtiene con las teclas SIMBOL SHIFT + A.

EL EDITOR «escribe» en la pantalla y además lo hace en una parte de la RAM, desde 4ØØØØ hasta 4ØØ73, es decir las 7Ø4 posiciones que maneja el basic (32*22=7Ø4) sin incluir las dos líneas de abajo.

Entre 4Ø8ØØ y 4Ø817, hay instalada una pequeña rutina que lee e imprime en pantalla los datos que se encuentren a partir de la posición 4ØØ18 hasta que haya un carácter 255, momento en el cual retorna a Basic.

Cuando se accede a la opción Compilar la rutina de compilación escribe a partir de la posición 4Ø818 los códigos ASCII del siguiente modo:

La primera letra de una palabra lleva 4 octetos los tres primeros indican la posición:

1. El 22, que es el código de AT (posición), por el que la rutina RST 1Ø de la ROM se «entera» que queremos imprimir en una parte concreta de la pantalla.
2. La fila donde ha de escribirse (de Ø a 21).
3. La columna (de Ø a 31).

El cuarto byte dice que carácter ASCII es la primera letra de la primera palabra.

El resto de las letras que forman la primera palabra sólo necesitan un byte por cada letra.

Los espacios en blanco son ignorados y sirven únicamente para indicarle al programa dónde empiezan y acaban las palabras, del mismo modo que nos lo indican a nosotros cuando leemos. No gastan por consiguiente, memoria. De este modo cada palabra ocupa tantos bytes como letras tenga + 3, que son los empleados para designar el lugar de colocación.

El sistema, aunque pudiera no parecerlo, ahorra memoria, pues en Basic, cada línea de program, cada instrucción PRINT PRINT AT, cada número gasta memoria, y los espacios en blanco también. En una pantalla llena se pueden ahorrar de 2ØØ a 25Ø bytes.

Al acabar la última letra de la última palabra se coloca un 255 para indicar a la rutina lectora el final.

Cuando ha sido compilado, el programa pregunta en que posición se desea colocar la rutina entre 42ØØØ y 63ØØØ; una vez introducida nos muestra la posición donde está instalado, la longitud en bytes y el último byte ocupado y nos ofrece la posibilidad de Ver lo realizado, volver al Editor o Salvar la pantalla en forma de bytes. Recomendamos una vez más tomar nota del primer byte de cada pantalla para poder llamarla después con USR.

La rutina colocada en 4Ø75Ø traspasa la pantalla desde 4Ø8ØØ a la posición que indique el usuario.

Terminado ésto, el programa borra desde 4ØØØØ a 4Ø7Ø3, escribiendo encima espacios (32), y vuelve a empezar otra pantalla.

**Precauciones y notas**

Al teclear el programa y dado que hay C/M y POKES delicados es conveniente salvar antes de hacer RUN.

Si ya se tienen compiladas algunas pantalla y se BREAKea el programa, es conveniente regresar con GOTO 6ØØØ pues si se hiciera con RUN no se borrarían las rutinas pero sí las tablas donde se archivan las direcciones donde están colocadas, es decir, no perdemos el «ovillo» pero sí el «hilo», que casi es lo mismo.

En la línea 99ØØ hay instrucciones para salvar el programa, basta por consiguiente hacer GOTO 99ØØ y se salvará con autoejecución en la línea 1.

## RUTINA DE COMPILACION DE PANTALLAS DE TEXTOS

```
10          ORG  #9A4C          ;decimal 39500
20          LD   BC,#9F72       ;puntero escritura 40818 d
30          LD   HL,#9C40       ;puntero de lectura 39999 d
40          LD   DE,#9C40       ;guarda inicio lectura 40000 d
50  INI     INC  HL
60          LD   A,(HL)         ;carga acumulador con el primer valor
70          CP   #FF            ;si es 255
80          JR   Z,FIN          ;se va a terminar
90          CP   #20            ;si es 32,un espacio,
100         JR   Z,INI          ;salta a inicio
110         LD   A,#16          ;si no es ni 255 ni 32,carga a=22 d
120         LD   (BC),A         ;almacena dicho valor en la posicion
130         INC  BC             ;que marca BC e incrementa BC
140         PUSH HL
150         SBC  HL,DE          ;reduce HL a un valor entre 0 y 703
160         PUSH HL             ;para averiguar sus coordenadas dentro
170         SRL  L              ;de la pantalla
180         SRL  L
190         SRL  L              ;divide L entre 32
200         SRL  L
210         SRL  L
220         BIT  0,H            ;pasa los bits 0 y 1 de H, si son 1,
230         JR   Z,SAL1         ;a las posiciones 3 y 4 de L
240         SET  3,L            ;con lo que queda completa la division
250 SAL1    BIT  1,H            ;por 32 de HL,para averiguar la
260         JR   Z,SAL2         ;posicion y (coordenada vertical) de
270         SET  4,L            ;la primera letra de una palabra
280 SAL2    LD   A,L            ;carga A con el valor obtenido
290         LD   (BC),A         ;lo almacena en posicion BC
300         INC  BC
310         POP  HL
320         RES  7,L            ;tacha los posibles 1 que tuvieran los
330         RES  6,L            ;bits 7,6,5 del registro L,para averi
340         RES  5,L            ;guar el resto de la division.
350         LD   A,L            ;este resto es la coordenada horizon-
360         LD   (BC),A         ;tal h de la primera letra que
370         INC  BC             ;almacenamos en posicion BC
380         POP  HL
390         LD   A,(HL)         ;toma el codigo ASCII de la primera
400         LD   (BC),A         ;lo coloca en posicion BC
410         INC  BC
420 COLA    INC  HL
430         LD   A,(HL)         ;lee el siguiente caracter
440         CP   #FF            ;si es 255
450         JR   Z,FIN          ;se va a FIN
460         CP   #20            ;si es 32 , incio
470         JR   Z,INI          ;se va al inicio
480         LD   (BC),A         ;si no es ni 255 ni 32 lo almacena
490         INC  BC             ;donde marque BC
500         JR   COLA           ;regresa a COLA para leer el siguiente
510 FIN     LD   A,#FF          ;al acabar coloca la marca 255 en el;
520         LD   (BC),A         ;ultimo BYTE de datos
530         INC  BC
540         LD   (#9F42),BC     ;pone el valor de BC el las posiciones
550         LD   C,B            ;de memoria 40770 y 40771 para que las
560         LD   (#9F43),BC     ;puede leer el BASIC
570         RET
```

# ¡NUEVO!

## SIEMPRE LOS PRIMEROS EN TENER LO ULTIMO

**circulo de soft**
MICROAMIGO S.A.

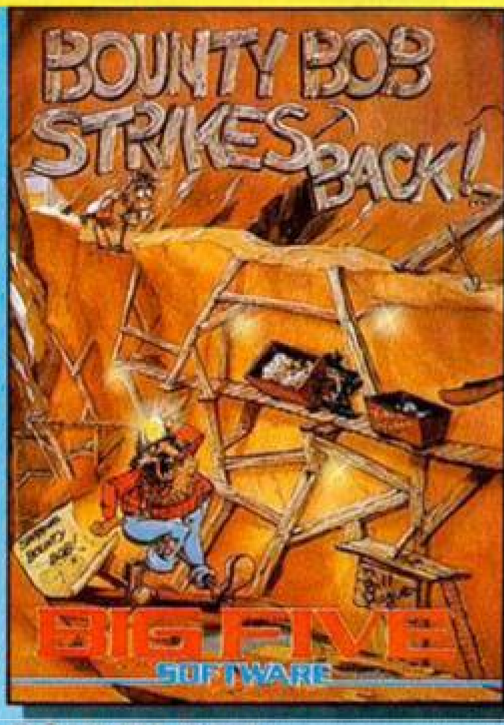
P.° de la Castellana, 268, 3.° C. 28046-MADRID.
Tel.: (91) 733 25 00




### FIGHTING WARRIOR

Para salvar a la princesa encerrada en la Gran Pirámide, tendrás que enfrentarte con tu espada a criaturas infernales que tratarán de impedírtelo por todos los medios. Este juego viene avalado por la firma de los creadores de Exploding Fist.

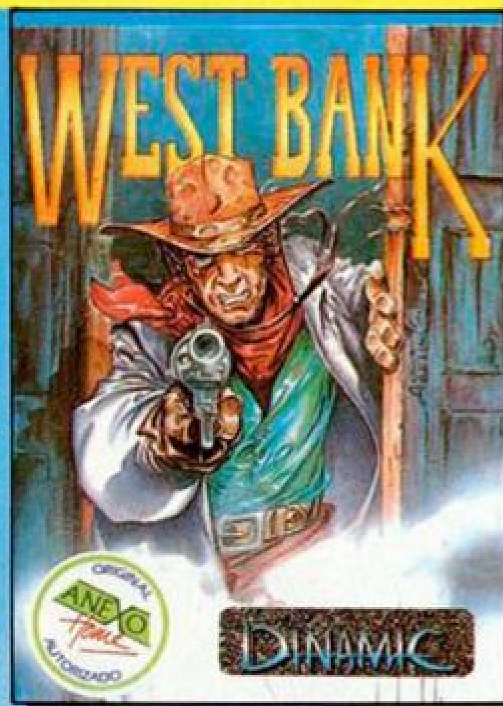
**P.V.P.: 2.100 ptas.**
**Precio Socios C. de Soft: 1.890 ptas.**




### BOUNTY BOB

Junto con el Manic Miner, éste es el mejor juego de «plataforma» aparecido para ordenador y cuyo éxito en Commodore va a repetirse ahora en su versión para Spectrum.

**P.V.P.: 2.100 ptas.**
**Precios Socios C. de Soft: 1.890 ptas.**




### WEST BANK

Defiende el banco de Soft City del ataque de los forajidos y consigue escribir tu nombre en la leyenda del «FAR WEST».

**P.V.P.: 2.100 ptas.**
**Precio Socios C. de Soft: 1.890 ptas.**

## ¡¡¡...Y LOS TRES PROGRAMAS POR SOLO 4.990 PTAS!!!

***¡HAZTE HOY MISMO SOCIO DEL CIRCULO DE SOFT!*** Además de poder adquirir tus programas al mejor precio, recibirás información de forma periódica y gratuita, del mejor software que aparezca en el mercado.

***¿QUE HAY QUE HACER PARA SER SOCIO DEL CIRCULO DE SOFT?*** Así de fácil: envíanos por correo tu nombre, dirección y modelo de ordenador, o bien, pide por teléfono o por correo tu primer programa. ¡Y entrarás a formar parte del CIRCULO DE SOFT de forma inmediata!

☐ SÍ, quiero ser SOCIO desde hoy mismo del CIRCULO DE SOFT y recibir periódicamente información de novedades de software, así como beneficiarme desde hoy mismo de los precios reducidos reservados a los SOCIOS y de sus Ofertas Especiales. El ser SOCIO no me obliga a compra alguna.

**Si prefieres formalizar tu compra por teléfono puedes hacerlo llamando al (91) 733 25 00. ¡¡NO SE COBRAN LOS GASTOS DE ENVIO POR CORREO!!**

| TITULO | P.V.P. | ORDENADOR |
|---|---|---|
| | | |
| | | |
| | | |

☐ Contrarreembolso ☐ Giro Postal ☐ Talón adjunto a Microamigo, S.A. ☐ Tarjeta VISA n.° ________ Fecha caducidad ________

Nombre ________ Apellidos ________ Edad ________

Domicilio ________ Teléfono ________

Localidad ________ C.P. ________ Provincia ________


DISPONIBLE PARA ZX SPECTRUM
AMSTRAD
SOFTWARE
Sound-on-Sound
Sound on Sound es una marca registrada
producida y distribuida por Iberofón, s.a.
Telf. 671.22.00 / 04/08/12/16
JUEGA CON EL FUTURO
THE ROCKY HORROR SHOW
SPECTRUM 48K
DON'T DREAM IT... PLAY IT!
RCA
ENTER AT YOUR OWN RISK
FREEZER TEMP
85
¡¡¡NO LO SUEÑES!!! ¡JUEGALO!
SIENTE LA EMOCION DE LO DESCONOCIDO
CORRE TU PROPIO RIESGO
SALVA A TU COMPAÑERO/A ATRAPADO/A
REUNE LOS FRAGMENTOS DEL CUADRO
SON TU AMULETO
¡¡¡POR FIN EN CASTELLANO!!!
LA PRIMERA COMEDIA MUSICAL EN VIDEO-JUEGO
Graphis, S.A Publicidad




# ALMACENAMIENTO DE PROGRAMAS EN MICRODRIVE

La primera operación a realizar obligatoriamente con un cartucho virgen es el formateado. Al igual que sucede con las cassettes de audio, es recomendable un «bobinado/rebobinado» del cartucho previo a su utilización, por medio del comando FORMAT, precisamente. Para ello, se aconseja escribir:

```
10 FORMAT "m";1;
   "nombre"
20 CAT 1
```

(suponiendo, obviamente, que manejamos el microdrive 1) y ejecutar el mini-programa con un RUN varias veces. Se observa, normalmente, que en 2 ó 3 formateados la capacidad del cartucho es superior a la obtenida la primera vez. Por otra parte, se sugiere que el «nombre» tenga relación directa con el contenido del cartucho (ej. JUEGOS1, UTIL3, etc.)

La segunda recomendación se refiere a la creación de un fichero llamado «run» (en minúsculas), que sirva a la vez de índice y cargador de, al menos, los programas más importantes contenidos en el cartucho, según se verá más adelante.

En tercer lugar, conviene recordar que, a diferencia de la cassette, no es posible hacer un "MERGE" de programas grabados en microdrive con auto-RUN (es decir, con SAVE... LINE), por lo que habrá que pensarse dos veces si se almacenan de esta manera programas que durante su ejecución no permiten el "BREAK", con vistas a realizar listados o futuras copias del programa.

Además, se recomienda que todos los "LOAD" de código máquina contenidos en programas almacenados en microdrive incluyan todos los parámetros además del nombre (que, por cierto, es obligatorio). Así, en lugar del parco LOAD "" CODE, escribiremos LOAD • "m";1; "nombre" CODE origen, longitud, con lo que el bloque de código quedará totalmente identificado para futuros usos.

Por último, se recomienda que toda operación de SAVE en microdrive vaya seguida por un VERIFY. En algunos casos aparecerá el fatídico "File not found" aunque el archivo sí figura en el catálogo. En estos casos, convendrá borrarlo con un ERASE "m";1;"nombre" y grabarlo de nuevo. Una fuente de frecuentes errores es la introducción involuntaria de códigos o caracteres extraños en el nombre (por ejemplo, comandos de color), por lo que conviene ser cuidadoso al respecto.

### *PASO DE PROGRAMAS DE CASSETTE A MICRODRIVE*

Una de las inevitables frustaciones del usuario del SPECTRUM que acaba de comprarse unos flamantes Interface 1 y Microdrive es la de comprobar con horror que sus programas favoritos, de los que desea hacer una copia a microdrive para beneficiarse de su velocidad de acceso, o bien no funcionan tras la conversión o, en muchos casos, no sabe siquiera por dónde «hincarles el diente». En este artículo se pretende orientar al usuario en los procedimientos de conversión ya que, por desgracia, dicha operación no es un asunto tan trivial como parece.

Como ya se ha dicho, debido al desplazamiento y ocupación de memoria adicional que sufren los programas en las operaciones asociadas al microdrive (un total de 685 bytes), no existe un método general de paso de programas de cassette a microdrive, dado que el grado de incompatibilidad de éstos puede variar desde cero al 100%. No obstante, se pueden establecer 8 categorías, que implican otros tantos procedimientos a seguir, dependiendo de dicho grado de «incompatibilidad». Como primera herramienta de trabajo es imprescindible un buen lector de cabeceras (listador), como el publicado en el n.º 2 de MICROHOBBY, para conocer las características del programa a convertir.

La mayoría de los programas comerciales constan de varias secciones que deberemos tratar por separado y que pertenecerán a alguna de las categorías indicadas. Como regla general, se recomienda prescindir de la pantalla de presentación, ya que en la cassette sólo sirve para entretener durante los minutos de carga del programa, que se reducen a escasos segundos con el microdrive. (Por otra parte, los cartuchos son más caros y tienen menos capacidad que la cassette).

Asimismo, conviene recordar que no es posible almacenar dos (o más) programas con el mismo nombre en el mismo cartucho, aunque sean de distinto tipo (BASIC o código máquina), por lo que en muchos casos habrá que cambiarles el nombre para almacenarlos. Se sugiere, a efectos de identificación, la introducción del carácter "CODE" (en modo extendido) en el nombre de los ficheros que sean código máquina.

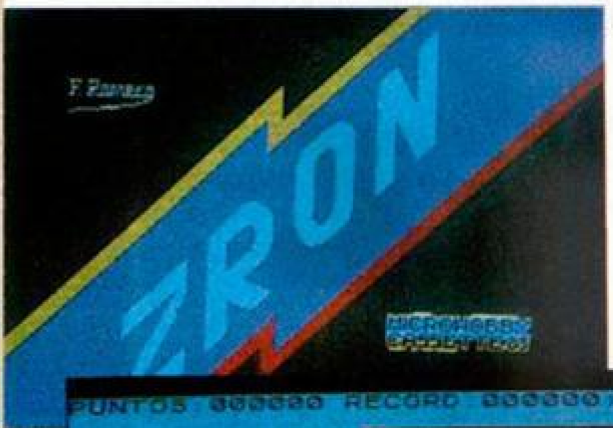
## METODOS DE CONVERSION

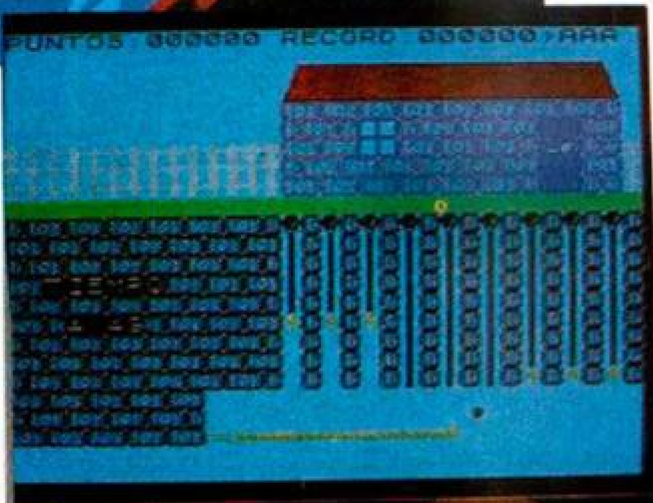
Los programas susceptibles de ser almacenados en microdrive se agrupan en las siguientes 8 categorías:

a) Programas totalmente en BASIC.
b) Programas en BASIC conteniendo código máquina en sentencias del tipo 1 REM...
c) Pantallas de presentación.
d) Código máquina en origen superior a 25000.
e) Código máquina con origen inferior a 25000 y longitud menor que 40500.
f) Código máquina con origen inferior a 25000 y longitud mayor que 40500.
g) Código máquina sin cabecera.
h) Programas «turbo».

# MICROHOBBY CASSETTE

## ¡YA EN TU KIOSKO!



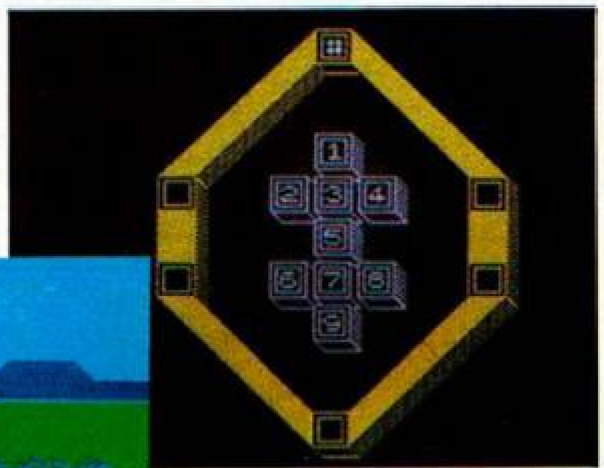

- 10 super programas para tu Spectrum (más de 200 K en total).
- Opción Joystick Sinclair o Kempston en los programas de acción.
- Esmerada programación Basic/Código máquina con excelentes gráficos.
- Instrucciones en 16 páginas a todo color.
- Garantía total de carga.
- Con el aval de MICROHOBBY.







**RECUERDALO... ¡NO TE QUEDES COLGADO!**







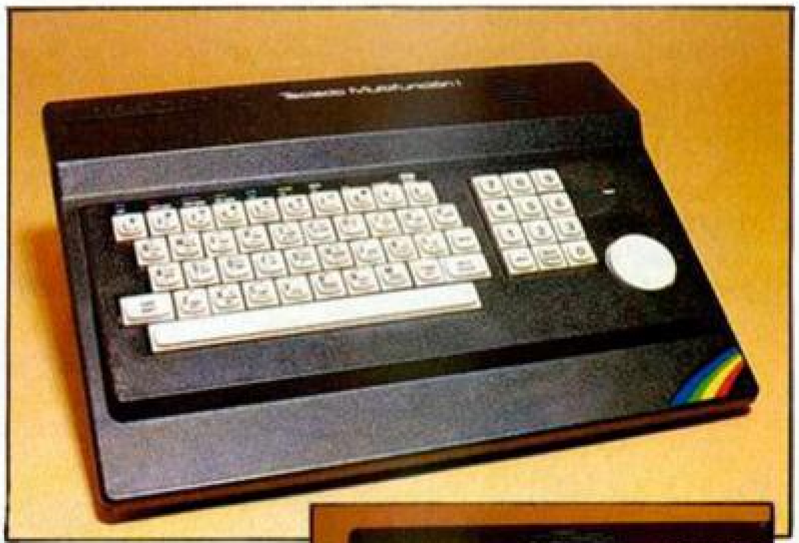



MICROHOBBY CASSETTE

**Sólo 395 ptas.**



# NUEVO TECLADO PROFESIONAL DE INDESCOMP





La nueva versión incorpora botón de reset, amplificador de sonido con volumen e interruptor ON-OFF.

Indescomp, a través de su distribuidor exclusivo LSB, acaba de lanzar al mercado un nuevo teclado profesional multifunción que presenta sustanciales mejoras respecto del modelo anterior.

Se trata de un teclado compacto que integra en su interior una serie de circuitos que amplían las prestaciones del Spectrum y es capaz de albergar en su interior tanto al Spectrum 48 K como a éste con el Interface 1 incluido. En ambos casos, la conexión se efectúa valiéndonos del Slot trasero, lo que evita el que tengamos que abrir o desmontar el Ordenador. Esto presenta la enorme ventaja de que la garantía no queda invalidada.



Las nuevas teclas son mucho más robustas.

Las teclas han sido sustituidas en su totalidad por otro modelo mucho más robusto y de tacto más agradable. Las teclas antiguas no daban la sensación de ser pulsadas, por lo que no se podía escribir cómodamente si no era consultando la pantalla continuamente para ver si se ha pulsado bien una tecla o no.

En las nuevas se nota claramente cuándo se ha pulsado correctamente.

También se ha cambiado la disposición de la tecla ENTER y ESPACE en el teclado numérico auxiliar.

El único "fallo" o despiste que hemos encontrado es que la salida de vídeo no lleva cable apantallado.

A continuación podéis ver un pequeño análisis comparativo entre los dos modelos que os permitirá haceros una idea clara de las mejoras introducidas.



El teclado anterior no tenía el amplificador integrado y el circuito impreso era de bakelita.

| Teclado Antiguo | Nuevo teclado |
|---|---|
| Sin filtraje en la alimentación. | Condensador de filtro (33ØØ microfaradios) con lo que la alimentación es más estable. |
| Decodifica la dirección de acceso mediante diodos conectados a las líneas A1, IORQ, RD. | Decodifica la dirección mediante el Circuito Integrado 74LS32. |
| Sin salida de vídeo. | Salida de Vídeo Compuesto. |
| Sin botón de Reset. | Con botón de Reset incorporado. |
| Amplificador de sonido a transistores. | Amplificador de sonido con el Integrado LM 386. |
| Placa de Circuito Impreso en bakelita y con carios puentes (1Ø) | Placa de Circuito Impreso de Fibra de Vidrio (mayor rigidez mecánica) y con diseño optimizado (sin puentes ni cableado extra). |
| Ambos modelos disponen de LED indicador de funcionamiento. | |

# CONSULTORIO

## MICROMANIA

*Poseo un AMSTRAD CPC 464 de 64K que empleo para estudiar programación de ordenadores.*

*Que yo sepa, no existe ninguna publicación semanal ni mensual, que se dedique a esta clase de ordenador y debido a esto, me encuentro con una falta de programas escritos.*

*Si saben de alguna publicación semanal, quincenal o mensual les agradecería que me dijeran dónde poder conseguirla.*

**Vicente PEREZ - Zaragoza**

□ Precisamente para usted y para quienes se hayan en su caso, hemos creado la revista MICROMANIA que encontrará puntualmente cada mes en su quiosco. En ella tratamos sobre todos los micro-ordenadores del mercado, incluido, por supuesto, el Spectrum.

## Avería

*Hace un mes que me compré el Plus, y hace unos días que no me sale el mensaje de (c) Sinclair... sino unas rayas de color rojo por toda la pantalla, y el teclado no responde. ¿Qué clase de avería tiene? ¿Puede arreglarse? ¿Cuánto costaría?*

**Rafael GALVEZ - Baleares**

□ Evidentemente, su ordenador sufre una avería. Desde aquí es imposible determinar de qué se trata. Le recomendamos que, dado que aún está en período de garantía, lleve el ordenador a reparar cuanto antes.

## Mantenimiento de tensión

*¿Qué le pasaría al ordenador si por casualidad al conectarle unas pilas lo hiciera al revés? La pregunta se debe a que me he hecho un aparato para que cuando se va la luz, el ordenador siga funcionando. Mantiene la tensión durante unos diez minutos. Si a algún lector le interesa hacérselo, es económico, que me avisa y le mando un esquema del circuito.*

**M. A. FERNANDEZ - S. Sebastián**

□ Ante todo, le agradecemos su ofrecimiento en nombre propio y de nuestros lectores. En cuanto a su pregunta, existen muchas posibilidades de que el ordenador sufra un daño irreversible si se conecta la polaridad invertida. Hay un circuito muy utilizado en emisoras de radioaficionado, que consiste en colocar un fusible en serie y un diodo en paralelo, que normalmente quede polarizado en inverso; si por alguna causa, se invirtiera la polaridad, el diodo conduciría haciendo saltar el fusible, y protegiendo el ordenador.

## En busca del programa perdido

*En el número 22 de vuestra revista, en el Artículo: «Construye tu propio Joystick II» decís que para utilizar el joystick se puede usar el programa que se publica en el número, pero no hay ningún programa con ese nombre. ¿Cuál hay que utilizar?*

**Jesús CHAMIZO - Madrid**

□ Por la existencia del mencionado artículo, fue necesaria su publicación en dos partes. El programa a que hace referencia, se encuentra en la página 24 del número 21.

## Réplica de un radioaficionado

*Al recibir la revista número 42, me he sorprendido muchísimo al leer en la sección Consultorio la respuesta que le dan al amigo Javier Fuentes de Cantabria respecto a que no se pueden transmitir programas por la Banda Ciudadana.*

*Les puedo informar a ustedes, y por supuesto a Javier, que es perfectamente posible, ya que nosotros lo venimos haciendo hace algún tiempo, tanto en AM como en FM y sin modem. La única condición es sustituir el micrófono por el cable de MIC para enviar el programa, y conectar el cable de EAR en el conector trasero de auricular que tienen la mayoría de emisoras de 27 MHz.*

*Para más detalles, les diré que mi equipo es una President Grant y hemos estado transmitiendo programas (y recibiendo) con equipos Master y Sommerkamp; si bien es cierto que el QRM puede estropear la transmisión y en ese caso hay que volver a repetirla, en el caso nuestro, al vivir en una población pequeña donde las distancias no son muy grandes (4 km.), la fuerza de la señal es suficiente para anular el QRM (S 9 + 20).*

*Creo siceramente que antes de asegurar tan categóricamente algo, hay que haber probado experimentalmente.*

*Sin otro particular, y una vez que me he desahogado, les saludo atentamente.*

**A. MARTORELL - Burgos**

□ La reducida banda pasante de la modulación en 27 MHz, unido al abundante QRM (Interferencias atmosféricas y de otras emisoras) que sufrimos en Madrid, hizo imposible obtener resultados positivos, en las pruebas que el encargado de esta sección (entusiasta de la Banda Ciudadana), realizó con una Major Multimode como equipo transmisor, y una Herton como equipo receptor.

Aunque nunca afirmamos nada categóricamente sin haber hecho antes abundantes pruebas, en ningún caso nos ha agradado tanto equivocarnos. Agradecemos al colega ECB 1 1 10008 su aclaración, y animamos a otros lectores a que experimenten y nos cuenten sus resultados.

## Permiso concedido

*Desearía saber si me dais permiso para utilizar la rutina en C. M. del sintetizador de voz que publicasteis, creo, en los números 20 y 21, para fusionar con un programa de utilidad en Basic que estoy realizando para vuestra revista.*

*Si la respuesta es afirmativa, ¿podría mandar el programa en Basic con la rutina en una cinta, o es necesario enviarlo con la rutina en líneas DATA?*

**Marcos Vidal - Cádiz**

□ Las rutinas que publicamos en nuestra revista tienen la finalidad de ayudar a nuestros lectores en sus programas, por tanto, están a su disposición siempre que no las usen con fines comerciales.

Si le es posible, preferimos que nos envíe su programa con la rutina DATAs, para facilitar el trabajo a aquellos de nuestros lectores que no la tengan.

## Código Máquina

*¿Cuál es la instrucción o rutina equivalente en C. M. a la instrucción Basic PRINT PEEK dirección?*

*¿Cómo puedo crear números aleatorios en Código Máquina?*

**Juan R. QUEROL - Madrid**

☐ No es posible establecer una equivalencia directa entre las instrucciones en Basic y en Código Máquina, no obstante, es perfectamente posible leer el contenido de posiciones de memoria desde Código Máquina; de hecho es la instrucción más fracuente. En el capítulo 5 de nuestro curso de C/M se muestra una forma de sacar el contenido de una dirección de memoria a pantalla.

En cuanto a la generación de números aleatorios, puede utilizar el registro "R" del microprocesador, o bien la variable FRAMES del sistema operativo. Esta y otras cuestiones serán tratadas más ampliamente en sucesivos capítulos del mencionado curso.

**Capacidad de memoria**

*1. El Microdrive ¿trabaja ON LINE?*

*2. ¿Cuántas líneas de pantalla (32 caracteres) caben en la memoria de un Spectrum de 48K, suponiendo que no hay almacenado ningún programa o que éste es muy corto?*

*3. Con todas las cintas que Vdes. me han mandado me ocurre lo mismo, sólo las puedo cargar por una cara, por la otra me da el temible "R TAPE loading error", ¿porqué puede ser?*

*Federico BAEZA - Ferrol*

☐ Hablar de funcionamiento ON LINE en un sistema reducido como el Spectrum, resulta un tanto ambiguo. En este ordenador, lo único que trabaja realmente resto de los dispositivos tra- ON LINE es la pantalla, el bajan de modo selectivo, de forma que cuando el microprocesador está atendiendo al Microdrive no puede atender a ninguna otra tarea.

La memoria disponible del Spectrum 48K, es de aproximadamente 41K, es decir 41.984 bytes; si dividimos este número por 32 caracteres, salen unas 1.312 líneas; si a su vez dividimos entre 22 líneas por pantalla, serían 59,6 pantallas de texto.

Respecto al problema de las cintas, puede deberse a un desajuste de cabezas, de forma que, aunque las de nuestro equipo estén dentro de tolerancia (a otros lectores no les ocurre ese problema), este pequeño desajuste, sea incompatible con el de las de su equipo. Es posible que pueda resolver el problema reajustando la cabeza reproductora de su cassette.

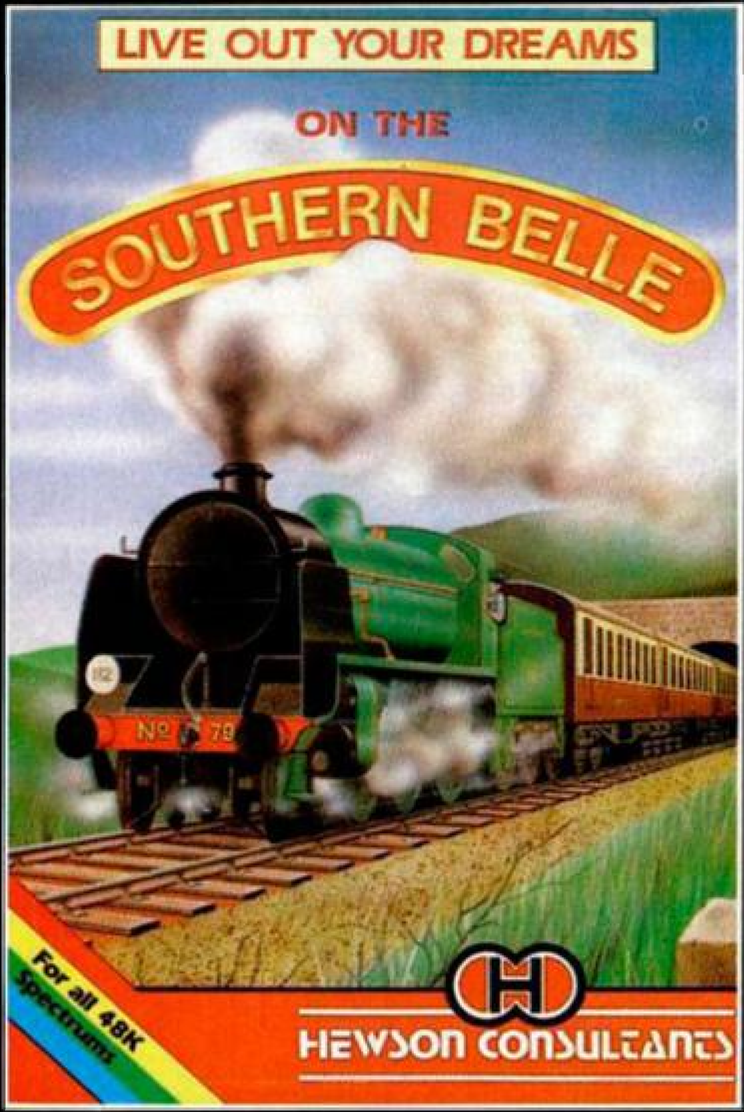
**Impresoras específicas**

*¿Las impresoras Seikosha-GP 50S y ZX-Printer necesitan de Interface? De ser así ¿qué modelo necesitan?; otra pregunta más, ¿la ZX Printer copia dibujos?*

*José M. LLAMAS - Valladolid*

☐ Tanto la Seikosa-GP 50S, como la ZX-Printer no necesitan interface, se conectan directamente al «slot» de expansión del ordenador.

Ambas pueden reproducir dibujos, mediante el comando COPY.



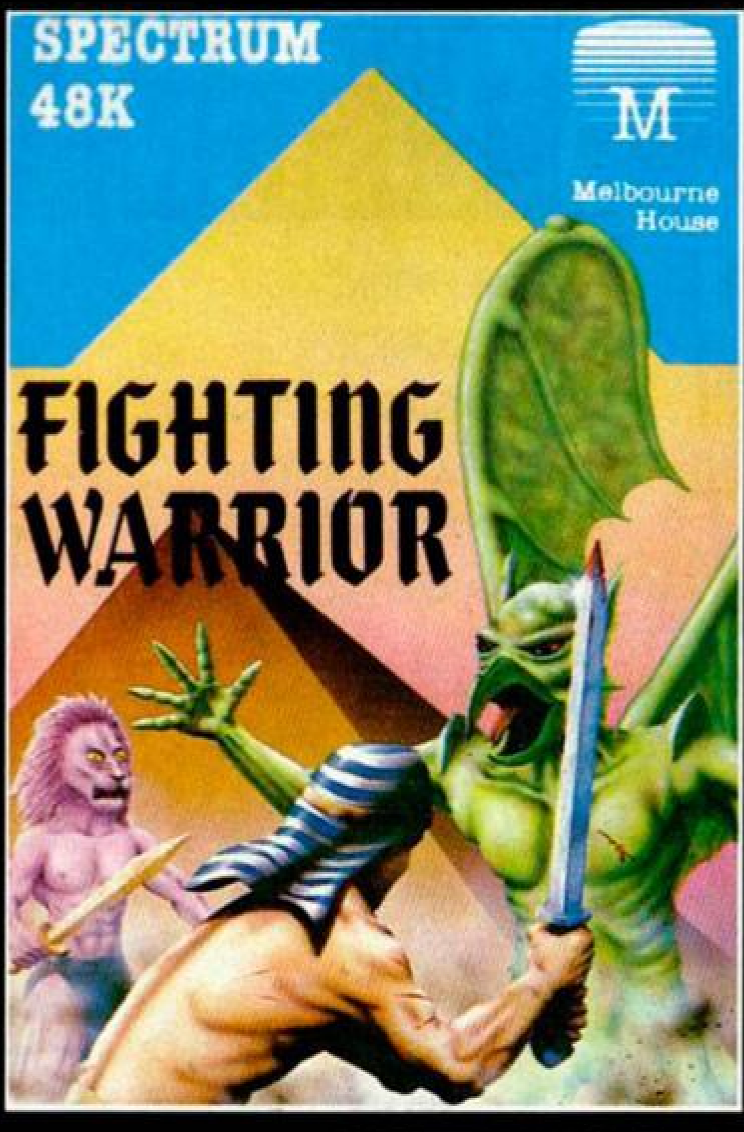

**SOUTHERN BELLE**

BIENVENIDOS A BORDO, OCUPEN SUS ASIENTOS Y PREPARENSE PARA UN EMOCIONANTE VIAJE EN "LA BELLA DEL SUR" LA MARAVILLOSA LOCOMOTORA DE LOS AÑOS 20.
CON ESTE PERFECTO SIMULADOR, CONTROLARAS TODOS LOS APARATOS DEL TREN, LA PRESION DEL VAPOR, EL CARBON DE LA CALDERA, LAS SEÑALES QUE ENCUENTRES EN LAS VIAS Y HASTA EL SILBATO QUE TENDRAS QUE USAR EN LOS MOMENTOS OPORTUNOS.
CONTROLAR UN TREN NO ES COSA FACIL Y PARA AQUELLOS QUE OS GUSTEN LOS SIMULADORES DE VUELO, OS VENDRA BIEN PONER LOS PIES EN TIERRA Y APRENDER A MANEJAR LA LOCOMOTORA QUE MARCO TODA UNA EPOCA.
SPECTRUM/AMSTRAD

Soft

¡TIE
LO
EXIT






**FIGHTING WARRIOR**

COMO FIGHTING WARRIOR CAMPEON DEL ANTIGUO EGIPTO, DEBES USAR TODOS TUS PODERES Y TECNICAS DE COMBATE EN EL DESAFIO QUE TENDRA COMO PREMIO EL RESCATE DE LA PRINCESA ENCERRADA EN LA GRAN PIRAMIDE. PARA LLEGAR A ELLA HABRAS DE MANTENER DUROS COMBATES CUERPO A CUERPO ARMADO SOLO CON TU ESPADA, QUE DEBERAS MANEJAR DE FORMA EXPERTA CONTRA CRIATURAS MITAD HOMBRE, MITAD DEMONIO QUE SURGIRAN, A TU PASO. SENSACIONES GRAFICOS Y MOVIMIENTOS CREADOS POR LOS MISMOS, QUE HICIERON "EXPLODING FIST" UNIDOS A UNA GRAN CANTIDAD DE SORPRESAS HARAN DE FIGHTING WARRIOR TU JUEGO PREFERIDO.
SPECTRUM.

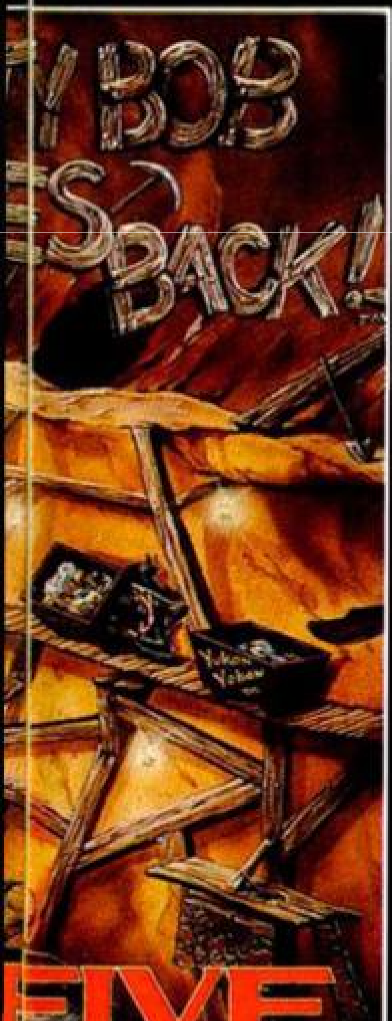
**BOUNTY BO S**

SI TE GUSTAN C
"PLATAFORMA" (BI
MIN
NO LO DUDE
ES TU
NO TE DECIMOS LA
FAVORITA Y DILES
¡SEGURO QUE

**DISTRIBUCION EXCLUSIVA PARA ESPAÑA ERBE SOFTW R**

ware

# ENE
# OS
# TOS!

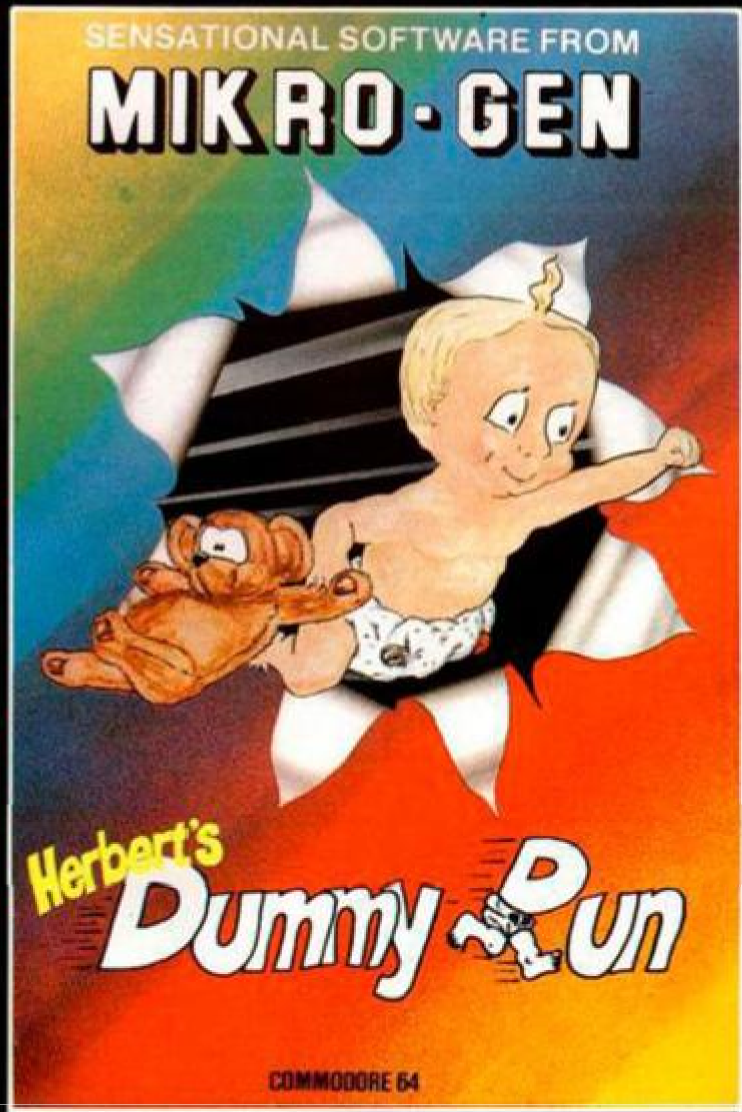



**E STRIKES BACK**

LOS JUEGOS DE
ABU SIMBEL, MANIC
ER, ETC.)
ES... BOUNTY BOB
JUEGO.
MAS, VE A TU TIENDA
QUE TE LO ENSEÑEN,
QUE TE GUSTA!

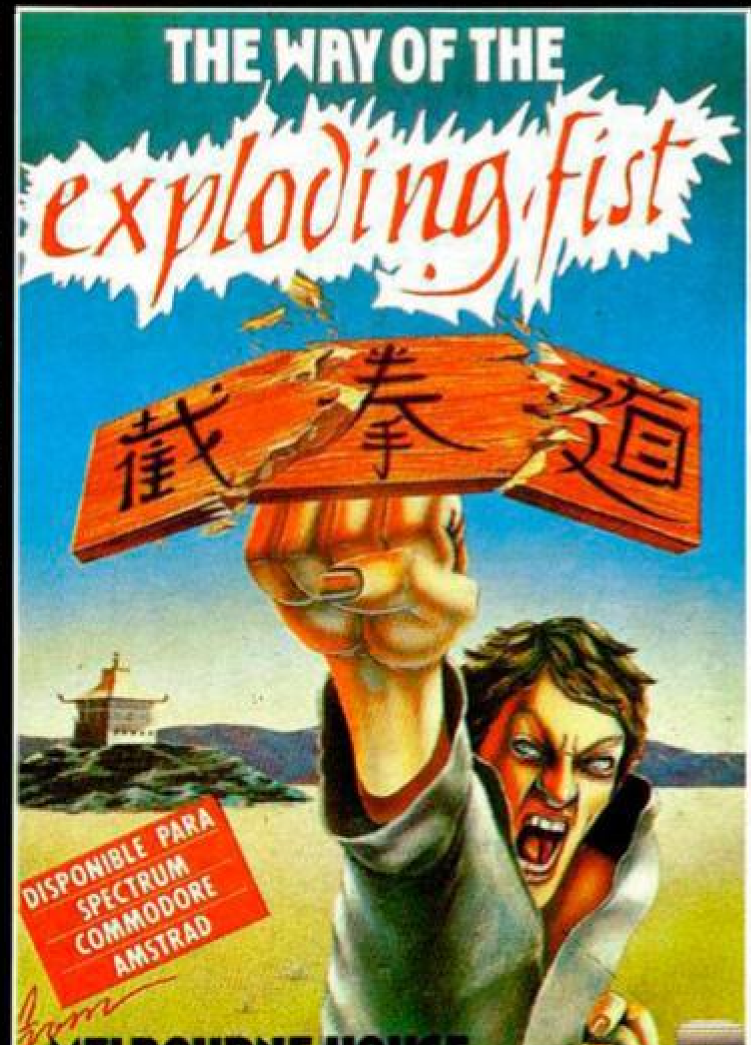
**HERBERT'S DUMMY RUN**

LA CRITICA HA DICHO: HERBERT'S ES UNA MEZCLA ENTRE EL "PYJAMARAMA" Y EL "EVERYONE'S A WALLY" DE LOS QUE HA ESCOGIDO LO MEJOR PERO AUMENTANDO LA CALIDAD DE GRAFICOS Y MOVIMIENTOS. RECOMENDADO PARA TODO TIPO DE USUARIOS, LOS QUE PREFIERAN LOS JUEGOS SENCILLOS Y LOS QUE QUIERAN JUEGOS UN POCO MAS COMPLICADOS EN LOS QUE ADEMAS DE SER HABILIDOSO QUIERAN ACTUAR CON LOGICA Y CREAR UNA ESTRATEGIA A SEGUIR. Y SOBRE TODO RECOMENDADO PARA LOS QUE DISFRUTARON CON PYJAMARAMA Y WALLY
SPECTRUM / COMMODORE / AMSTRAD




**EXPLODING FIST**

EL MEJOR JUEGO DE KARATE QUE PUEDAS ENCONTRAR. EL PROGRAMA MAS VENDIDO ACTUALMENTE EN INGLATERRA. 18 GOLPES Y MOVIMIENTOS DISTINTOS CUIDADOSAMENTE DISEÑADOS PARA SIMULAR UN EMOCIONANTE COMBATE EN EL QUE TODOS TUS REFLEJOS Y HABILIDADES SERAN PUESTAS A PRUEBA.
SPECTRUM / COMMODORE / AMSTRAD




**ARE, SANTA ENGRACIA, 17, TEL. 447 34 10. 28010 MADRID**

# DE OCASION

● CAMBIARIA enciclopedia de Aeromodelismo y Radio Control que consta de 54 fascículos y 3 tapas de encuadernación y además 1 Kit de avión Escuela sin usar, por ampliación interna del Spectrum o bien lo vendo por 7.500 ptas. Tel.: (954)621093 (3,30 a 9). Sevilla.

● VENDO un Spectrum 48 K, con garantía, con todos sus accesorios, con «Reset» incorporado, manual en castellano y con una gran cantidad de información. Precio: 30.000 ptas. Tel.: 3359375 de Barcelona.

● DESEARIA que me enviasen las instrucciones del ensamblador GENS-3 en inglés o castellano. Pago gastos de fotocopias. Escribir a Juan José Caballero. Antonio Ulloa, 4. 4.°. Cádiz.

● VENDO ZX Spectrum Plus, comprado hace un mes aprox. Impecable y con garantía de un año, todo por 30.000 ptas. Escribir a José Serrano Feriole. Font Nova, 32. Sabadell (Barcelona).

● VENDO/CAMBIO por impresora para ordenador un radiocontrol de 6 canales, 4 servos (con velero, avioneta a motor de 3,5 c/c y lancha rápida, etc.) Tel.: 579673. Asturias.

● VENDO Spectrum 48 K, con botón de Reset y salida de vídeo compuesta. Precio: 27.000 ptas. Interesados escribir a Enrique Novella Casado. Sta. Susana, 4. Madrid.

● VENDO ZX 81 con amplificador a 16 K, inversor de vídeo y pulsador de Reset incorporado, salida de conectores para caja de nueve pulsadores. Regalo la caja de alimentación estabilizada, manual y dosier completo, revistas, libros, etc. Todo por sólo 14.000 ptas. Llamar al Tel.: (94)4451377. Preguntar por Daniel.

● QUISIERA que me mandasen las instrucciones del programa Underwulde. Pagaría gastos de envío y la fotocopia. Colegio Guardias Jóvenes, Bl. 1, 6.° D. Valdelmorso (Madrid).

● HACEMOS pantallas de presentación (Sreen normal o instantáneo). Hacemos certificados de rendimiento, programas en Basic y rutinas en C. M. Escribir a Eduardo García López. Avda. Valladolid, 19, 7.ª. Palencia.

● DESEARIA contactar con usuarios del ZX Spectrum únicamente de la provincia de Gerona, para formar un club. Escribir a Luis Ribas. Ctra. Sagaro, s/n. Castillo de Aro (Gerona).

● COMPRO ordenador 48 K, no importa en las condiciones en las que esté, o bien lo cambio por video-juego con diez tipos de deportes. Interesados escribir a Miguel Martín Hornillos. Bda. Sta. Ana, 44. Nerva (Huelva).

● DESEARIA intercambiar trucos, ideas, información, etc. para el Spectrum 16/48K. Escribir a José M.ª Chacon Losada, Bda. Buen Pastor, 317. San Fernando (Cádiz). Tel.: 893557.

● VENDO ordenador ZX-81 de 32 K de memoria por 16.500 ptas. Interesados escribir a la siguiente dirección: Manuel Soler Llorca. Urb. Azmirra, 8. Campo de Mirra (Alicante) o bien llamr al Tel.: 801527 (10 a 12 mañanas).

● BUSCO impresora para ZX Spectrum. Interesados contactar con Antonio López Sánchez. Daniel Gil, 51, 3.°. Onteniente (Valencia). Tel. (96)2385348.

● DESEO entrar en contacto con usuarios del Spectrum. Mi dirección es: Altamira López de Silanes. Alfonso VI, 17, 4.°. Miranda de Ebro (Burgos).

● VENDO consola de videojuegos Atari video Computer Sistem 2600 TM, con cables, transformador, mandos de joystick y de paleta, con 5 cartuchos. Todo por 150.000 ptas. Llamar al Tel. 2002013 de Madrid. Preguntar por José Miguel.

● DESEARIA contactar con chicos que tengan un Spectrum 16/48K. Interesados escribir a la siguiente dirección. Miguel Facal, Avda. de Cádiz, 8, 2.° C. Granada.

● CAMBIO juego de tres pantallas, casi nuevo: "Rabbit Hop", valorado en 7.000 ptas. Tienes relojes independientes entre sí, día, mes, cronómetro, alarma e interruptor de sonido. Lo cambio por joystick con interface. Si te interesa llama al Tel. (927)411111 (14 a 18 horas). Preguntar por Víctor.

● VENDO amplificaciones de sonido para Spectrum. Potencia de 2 W. y además un reset incorporado. Precio: 3.000 ptas. Envíos contrareembolso. Salvador Beltrán. San José, 10. Córdoba 14009.

● VENDO/CAMBIO, ordenador ZX Spectrum Plus 64 K, completo, en periodo de garantía con 6 video-juegos de regalo y un interface y su correspondiente joystick Quick Shot II, por sólo 35.000 ptas. Lo cambiaría por un vídeo VHS o Beta o un TV en color de 14". Todo ello en funcionamiento. Daría facilidades de pago a los residentes en Barcelona. Interesados llamar al Tel.(93)2115849 de Barcelona.

● DESEO contactar con "Sam & P.Soft" de Cádiz para intercambio de ideas. Escribir a Luis Yanes Melús. S. Juan (Marina), blo. 4, bj. Ctro. Puerto de Sta. Mª. Cádiz o llamar al Tel. 850931. Preferiblemente escribir.

● DESEO tomar contacto con algún club de Alicante. Escribir a: Agustín Martínez Valera. Avda. Constitución, 14, 3.° A. Villena (Alicante).

● VENDO VIC 20, más 4 libros de programación, 9 revistas por 20.000 ptas. o bien lo cambio por Spectrum 48/16K. Interesados escribir a Gonzalo Bermúdez Veiras Lestedo. Santiago de Compostela (Coruña). Tel. 503076.

● VENDO lote incluyendo Spectrum 48K, con garantía Investrónica, cables, adaptador, manuales en inglés y castellano, botón Reset, cinta de demostración, revista del ramo, cintas vírgenes, plantilla, listados, información. Todo por 28.500 ptas. Vendo interface programables Indescomp sin estrenar, con instrucciones y garantía (6 meses). Precio: 38.000 ptas. Interesados llamar al Tel. (982)217979 o bien escribir a Eduardo Vilanova Arias. Yañez Rebolo, 99, 3.°. Lugo.

● HEMOS formado un club de usuarios del ZX Spectrum. Es un club diferente. Para todas las edades. Escribir o llamar a Angel Durán García. Plaza Fonsagrada, 5. Madrid. Tel. 2016602.

● AQUELLAS personas que residan el S. Pedro, Marbella, y Estepona, que deseen intercambio de Software, que se pongan en contacto conmigo, llamando al Tel. (952)780086 y preguntar por Pedro. También intercambio con toda España, dirigiéndose por carta a la siguiente dirección: Jorge Romero Mena. López de Mena, 20, S. Pedro de Alcántara (Málaga).

● VENDO ZX Spectrum 48K, más interface Joysctick Kempston con lápiz óptico. Contactar con Juan José Tel. 2019105 de Madrid. Regalo de libros y revistas. Precio: 25.000 ptas. (negociables).







## ACLARANDO QUE ES GERUNDIO

Se habrán extrañado nuestros queridos lectores al encontrar un prólogo, de improviso, en el curso de Código Máquina de esta semana. Pero todo tiene su explicación y por supuesto, sus motivos, y el nuestro no ha sido otro que el de dar una respuesta a el gran número de cartas recibidas en la redacción planteando diferentes cuestiones: para qué sirve, ventajas sobre el BASIC, alto nivel del Código Máquina ofrecido... Por lo que nos hemos creído en la obligación (y porque más vale tarde que nunca) de insertar un prólogo, con una paginación diferenciada, en el que quede perfectamente aclarada la intención que nos ha llevado a la publicación de este interesante curso que intenta dar una perfecta y clara visión de este difícil lenguaje.

En este capítulo, además, se ofrecen unos conceptos básicos de Código Máquina para todos aquéllos que no hayan conectado bien con el curso y, en definitiva, enlazar a nuestros «seguidores» de BASIC con la complejidad del Código Máquina.


SPECTRUM
128
INVESTRONICA
POWER
128K